NAGW-4435

# DEVELOPMENT OF A COMPUTER MODEL TO PREDICT PLATFORM STATION KEEPING REQUIREMENTS IN THE GULF OF MEXICO USING REMOTE SENSING DATA

Submitted to:

Mr. James Aliberti
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Kennedy Space Center
Florida

Prepared by:

Bryan Barber, Team Leader
Laura Kahn
David Wong

Mechanical Engineering Department
THE UNIVERSITY OF TEXAS AT AUSTIN
Austin, Texas

April 9, 1990

(NASA-CR-186678) DEVELOPMENT OF A COMPUTER MODEL TO PREDICT PLATFORM STATION KEEPING REQUIREMENTS IN THE GULF OF MEXICO USING REMOTE SENSING DATA (Texas Univ.) 124 p CSCL 22A G3/13

N90-25155

Unclas
0289181

MECHANICAL ENGINEERING DESIGN PROJECTS PROGRAM
THE UNIVERSITY OF TEXAS AT AUSTIN

*ETC 4.102 · Austin, Texas 78712-1063 · (512) 471-3900*

April 9, 1990

Mr. James Aliberti
Mail Stop PT-PMO
Kennedy Space Center
Florida 32899

Dear Mr. Aliberti:

Attached is our final project paper for simulating conditions encountered by a semisubmersible platform in the Gulf of Mexico.

The Mechanical Engineering Design Projects Program will hold oral presentations from April 23 to April 25, 1990. Our presentation will be Tuesday, April 24, at 8:00 am. Also, a catered luncheon will be provided on the day of the presentation for team members, sponsors, and faculty advisors. We hope that you can attend. We look forward to meeting you.

Sincerely,

Bryan Barber

Bryan Barber, Team Leader

Laura Kahn

Laura Kahn

David Wong

David Wong

## ACKNOWLEDGEMENTS

The team would like to thank Mr. James Aliberti and the National Aeronautics and Space Administration for sponsoring this project. Special thanks to Dr. Melba Crawford of the Mechanical Engineering Department at UT/Austin for advising the team and providing information for the project. Special thanks also to Dr. John Lundberg of the Aerospace Engineering Department at UT/Austin for serving as faculty advisor and also providing information for the project. And, thanks to Mr. Thomas Suniga, a research assistant in the Imaging Lab in the Mechanical Engineering Department at UT/Austin, for all his help in learning operating systems and gathering data.

The team would like to express its appreciation to Mr. Richard Connell for all his help and guidance throughout the course of this project. The team would also like to extend its thanks to Dr. Steven Nichols, Mr. Bert Herigstad, Mr Wendell Deen, and Professor Louis Torfason for all their time and assistance throughout the semester.

Special thanks to Hien T. Djie from Brown & Root U.S.A. in Houston, Texas, for his generous assistance in the area of vessel performance.

# ABSTRACT

DEVELOPMENT OF A COMPUTER MODEL TO PREDICT PLATFORM STATION KEEPING REQUIREMENTS IN THE GULF OF MEXICO USING REMOTE SENSING DATA

Offshore operations such as oil drilling and radar monitoring require semisubmersible platforms to remain stationary at specific locations in the Gulf of Mexico. Ocean currents, wind, and waves in the Gulf of Mexico tend to move platforms away from their desired locations. The team has created a computer model to predict the station keeping requirements of a platform. The computer simulation uses remote sensing data from satellites and buoys as input. A background of the project, alternate approaches to the project, and the details of the simulation are presented in this paper.

Bryan Barber

Bryan Barber, Team Leader

Laura Kahn

Laura Kahn

David Wong

David Wong

## TABLE OF CONTENTS

APPENDICES

## LIST OF FIGURES

Page No.

# INTRODUCTION

The Universities Space Research Association (USRA) coordinates a group of universities cooperating in the exploration and development of space. USRA was formed in 1969 by the National Academy of Sciences to further space research and technology. Based in Houston, USRA works under the guidance of the U.S. National Aeronautics and Space Administration (NASA), a national space and aeronautics agency established by the federal government in 1957.[1] These organizations sponsor space related research projects at The University of Texas at Austin (UT/Austin). USRA is interested in using remote sensing data to model the conditions in the Gulf of Mexico and has sponsored this project at the UT/Austin Mechanical Engineering Department.

## Background

Ocean circulation in the Gulf of Mexico is important to a wide range of industries including shipping, deep water exploration and production of oil and gas, and commercial fishing. Industries such as shipping and fishing require ships to move through the water, while oil drilling and drug interdiction efforts (such as the United States Navy's Deep Ocean Research Island (DORI) Project) require a dynamically positioned vessel to remain stationary. Dynamically positioned vessels (for example, semisubmersible platforms) are not anchored, but rather depend on positioning motors to keep them stationary.

Oil drilling efforts require a dynamically positioned vessel to maintain station in the deep waters of the Gulf of Mexico. The rate of ocean drilling depends largely on sea wave height and currents, which interfere with the vessel's ability to maintain station. Wave height varies with wind velocity. Ocean waves cause vertical translation, or heave, of the vessel. Because the drill pipe is connected to the sea floor, the vertical movement of the vessel can create a tension strong enough to break the pipe connections. The vessel's lateral movement must be maintained within three percent of the vertical distance to the ocean floor. [2] Excessive lateral movement of the vessel leads to horizontal shear forces in the pipe which can damage or break the connections.

The U.S. Navy also requires stationary vessels. The U.S. Navy proposes (in the DORI Project) to set radar balloons (Aerostats) in the Gulf of Mexico. These Aerostats will track low flying aircraft suspected of transporting drugs. These radar balloons will be tethered to a stationary vessel positioned for maximum radar coverage of the Gulf of Mexico.[3]

Ocean currents tend to move these vessels off station. The waters in the currents in the Gulf of Mexico can travel as fast as four knots. The Gulf Stream is the large scale ocean current which transports warm equatorial water through the Gulf and up the Eastern seaboard. The Gulf Stream is called the Loop Current within the Gulf of Mexico. Eddies are massive bowl shaped columns of rotating water spawned by the Gulf Stream which can also affect the vessel's ability to maintain station. These eddies can be up to 400 kilometers across and 500 meters deep.[3]

The water in the currents and warm core eddies is typically warmer than the surrounding water. The height differential between the warm and cool water can be as much as a meter, with the warmer water being higher. Both their temperature and height characteristics can be used to track the currents and eddies. Temperature readings are collected by U.S. National Oceanic and Atmospheric Administration (NOAA) satellites using Advanced Very High Resolution Radiometers (AVHRR). Figure 1 shows a NOAA satellite.

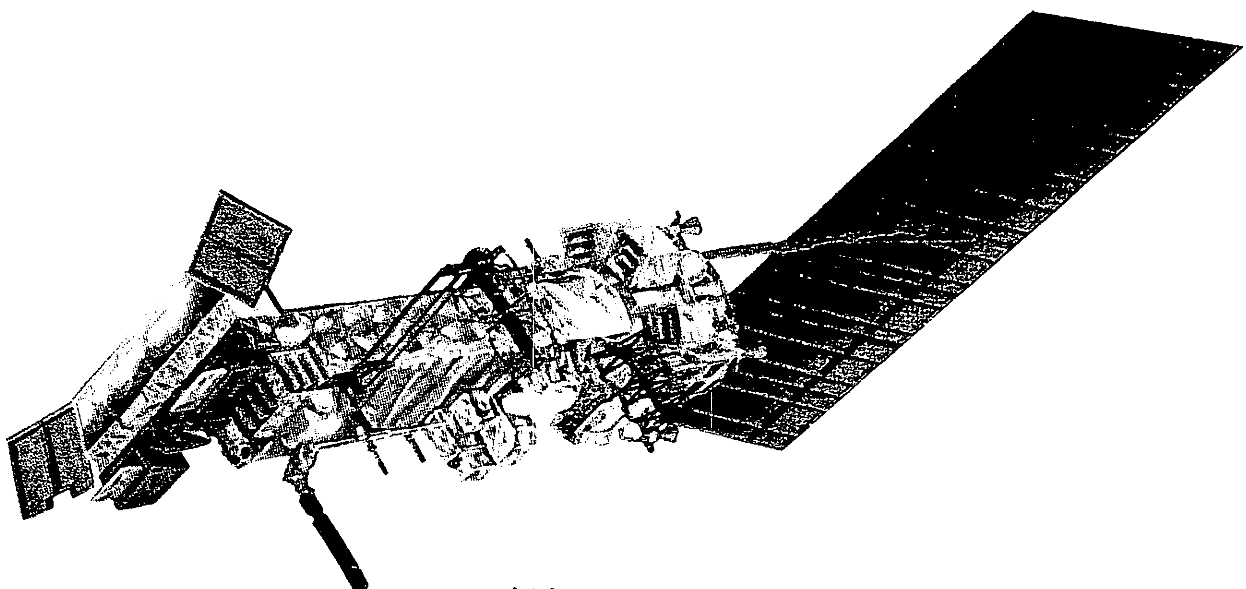



**FIGURE 1: A NOAA SATELLITE**

The most recent sea height readings were compiled by the GEOSAT mission. The GEOSAT mission ran from 1986 to January, 1990. This satellite mission carried a radar altimeter, which measured time for a transmitted radar pulse to travel from the satellite to the surface of the ocean and back. This time is used to measure the distance from satellite to surface, which can be used to calculate sea height if the satellite's position is known. No satellites with altimeters are currently in operation; therefore, the team will use previously compiled sea height data. Sea height measures include significant wave height, which can be used to estimate wind speed and ocean topography. Buoy transmissions will also be used to track the movements of the Gulf Stream eddies. [4]

Because current and eddy phenomena have not been accurately modeled, station keeping requirements (fuel, resupply, etc) cannot be reliably forecasted. A dependable simulation model is needed to predict the vessel station keeping requirements. The purpose of this project is to develop such a model.

The results of this project are important to USRA/NASA, the U.S. Navy, and various companies involved in offshore oil drilling operations. This project is a good example of how USRA/NASA's satellite remote sensing capabilities can be applied. The simulation's application of remote sensing is especially appropriate since NASA is interested in directing more of its efforts towards inner space (Mission to Planet Earth) rather than outer space. The U.S. Navy's interest in this project relates to the maintaining of a stationary platform in the Gulf of Mexico. The recently canceled Deep Ocean Research Island (DORI) project [5] was the original basis of this project, and any future radar surveillance projects will be able to use the results of the simulation. Oil companies interested in drilling in very deep water in the Gulf of Mexico can use the simulation's results to determine the viability of specific locations in the Gulf of Mexico for petroleum production.

## Project Requirements

The computer simulation model was developed from an existing FORTRAN code for Arctic Ocean drilling vessel simulations. The data input and parameters for the model were thermal data and altimeter readings from remote sensing satellites and flow vector information from buoys in the Gulf of Mexico. The thermal data was ocean surface temperature readings of the Gulf from Advanced Very High Resolution Radiometers (AVHRR) mounted on U.S. National Oceanic and Atmospheric Administration (NOAA) Satellites 9, 10 and 11. These radiometers sense the temperature of the top millimeter of the water. These NOAA satellites cover every point on the earth twice per day. Theoretically, this allows six images of a particular point (like the Gulf of Mexico) per day. Commonly, only three of the images are not overly distorted by the curvature of the earth's surface. The buoy transmissions were also collected by the NOAA satellites.

The team used the surface temperature readings to calculate temperature gradients. These gradients were used to locate the eddies and currents. The team then tracked the changes of these gradients over time to determine current movement. The team also correlated the temperature readings with spatially interpolated altimeter data in an effort to gain more accurate insight into patterns of currents and weather conditions. The altimeter readings were only taken directly under the satellite and the successive passes were far enough apart that they required interpolation to show a trend. A model was developed for wave conditions and another model was developed to predict storm arrival and severity. These models were used to generate data for the simulation. Development of an accurate computer simulation model was the final goal of this project.

## Project Criterion

The project criterion was to develop a working computer simulation model for a dynamically positioned vessel in the Gulf of Mexico.

The simulation model was to be tested by comparing its results for a previous time period to actual data from that period.

## Methodology

The design team proposed to address the Gulf of Mexico station keeping simulation project in four stages: general research, collection and processing of remote sensing data, modification of existing computer simulation model, and final compilation of input data to implement a working simulation model. The time available dictated the size of the data base which the team used as input for the computer simulation model.

In addition to written sources of information, each team member consulted closely with experts in his or her respective area of research. Dr. Melba Crawford, who helped to write the original computer simulation model, assisted in new program development and modification of the existing FORTRAN code for the model. The remote sensing data was comprised of surface temperature readings and sea height measurements from orbiting satellites. In addition, current movements were tracked by buoy transmissions. Mr. Thomas Suniga aided in the preparation of the thermal imaging data. Mr. Suniga is a research assistant in the Mechanical Engineering remote sensing lab. Dr. John Lundberg, of the Aerospace Department at UT/Austin, supplied sea height readings and wind speed estimates from the GEOSAT mission. All remote sensing data was taken from the time period of Spring 1989 because the weather conditions of that period allowed exceptionally clear thermal images to be obtained.

The thermal imaging data was used to track the Loop Current and the eddies it spawns in the Gulf of Mexico. Successive images were correlated using a program available at UT/Austin. [6] This program connects individual points on the images to their positions on subsequent images, giving the vectors necessary to calculate velocity and direction of current and eddy movements. Buoy transmissions were used to confirm these calculations by smoothing buoy point locations into trajectories. This "smoothing" was accomplished with appropriate curve fitting techniques.

Sea height data from the GEOSAT altimeter was used to calculate wave height and wind speed. Wave height is important because it causes the vertical motion, or heave, of the vessel. Wind speed is both an indicator of weather conditions (storms are classified by wind speed) and a contributor to wave height. Macroscopic sea height trends were also used to track the loop current, as the warm water in the current can be as high as a meter above cooler surrounding water. Spatial interpolation of these large scale readings was performed to track the movements of currents and eddies in addition to the thermal imaging and buoy tracking results.

The computer simulation model was based on an Arctic Ocean oil drilling simulation developed for ARCO by Susan Hoffman and Dr. Melba Crawford, both of UT/Austin. The new model incorporated subroutines from the Arctic model with new code developed specifically for the Gulf of Mexico. Initially, the computer simulation model only describes the vessel's station keeping requirements. This applies to any vessel attempting to keep station in the Gulf of Mexico. In continuing projects, drilling operations (with respect to time, resupply, weather, etc.) will also be included in the computer simulation model. This will expand the usefulness of the computer simulation model to oil drilling and exploration operations as well. The remote sensing data in its processed form was used to generate input data for the simulation.

Throughout the project, the team periodically consulted with Dr. Melba Crawford, Dr. John Lundberg, and Mr. Rick Connell to assist the team in meeting the project requirements. The team gave several practice presentations in order to gain a familiarity of the project material and speaking for an audience.

# ALTERNATE APPROACHES

This chapter presents three areas of flexibility in the team's computer simulation model process. These areas are:

1. Use of the existing model
2. Inputs to the model
3. Smoothing techniques for input data.

The team used an existing computer simulation model as specified by the team's project contact, Dr. Melba Crawford. The specific use of the model was flexible with respect to the inclusion of oil drilling operations and movement due to currents. Weather and time period were option areas to be considered for the data inputs. Finally, various curve fitting techniques were evaluated for use in fitting continuous functions to discrete data points.

## Use of the Existing Model

Dr. Melba Crawford specified the use of the existing model developed for the Arctic Ocean for the Gulf of Mexico simulation. The model consists of three main parts: the network model, the continuous event segment, and the discrete event segment.

The network model tracks the operations of the stationary vessel and the supply ships. This network model was written in SLAM, a simulation language. The network interacts with the discrete event segment over simulated time to process the activities of the simulated vessels as events occur. [2]

The continuous event segment models continuous functions such as weather, supply ship trips, and effective time spent on operations. This segment runs concurrently with the discrete event segment and the network. The continuous segment updates the state variables (such as supply levels and weather conditions) at the end of each time step. [2]

The discrete event segment updates discrete events as time or supply levels cross threshold values in the continuous segment. The discrete events modeled include beginning and ending of storms, supply ship arrival, and the end of the simulation. This segment stops the simulation when the ending time of the simulation has been reached or when there are no events remaining on the event calendar. [2]

The team determined how much of the original code was applicable to the new simulation. Subroutines directly related to oil drilling operations were either deleted from the program entirely, rewritten to include only station keeping activities, or included in original form.

The effects of ocean currents on a vessel trying to keep station required new simulation code. The team decided which computer programming language best met the requirements of the simulation. FORTRAN and SLAM were the two languages considered. FORTRAN is a readily known, all purpose programming language but lacks features such as discrete event simulation. SLAM is a self documenting and flexible language developed specifically for simulation purposes.

Inputs to the Model

Weather conditions and time period were two areas of simulation input which could be approached in more than one way. The weather conditions in the Gulf of Mexico could be estimated with data taken from Cape Hatteras, North Carolina, GEOSAT altimeter readings, National Weather Service and United States Navy barometric charts, or a combination of these three sources.

ARCO Oil and Gas Company has documented weather conditions in the waters off Cape Hatteras, North Carolina. [7] Cape Hatteras weather conditions are fairly similar to weather conditions in the Gulf of Mexico. If weather information for the Gulf of Mexico was not adequate for the simulation, the Cape Hatteras information could be used to predict the Gulf of Mexico weather in the simulation. [3]

Gulf of Mexico weather conditions could also be predicted by hindcasting wind speed from significant wave height readings taken by GEOSAT. If the wave height readings proved to be an accurate predictor of wind speed, then actual Gulf of Mexico weather conditions could be used in the model. These GEOSAT predictions could be checked for accuracy by comparing them to records of actual weather conditions at the time of the altimeter readings. [4]

Finally, barometric charts available from the National Weather Service and the U.S. Navy show the passage of weather fronts, which indicate wind direction and speed. [8] These charts could be an excellent source of information on the Gulf of Mexico weather conditions and could be used in conjunction with the altimeter data.

The time period of the simulation could range from two months to all twelve months of the year. The shorter period was required if the simulation proved too lengthy and complex. A lengthier simulation period included more seasonal changes in weather conditions. These seasonal changes increased the complexity of the weather forecasting required for the simulation.

The summer season brings uniformly warm temperatures and humid atmospheric conditions to the Gulf of Mexico, which lead to poor resolution in the thermal images. Spring, fall, and winter bring greater contrast in temperature between the Gulf of Mexico and the Loop Current. This increased temperature contrast allows much clearer thermal images to be obtained. If altimeter readings of macroscopic sea height proved unreliable in tracking the movement of the Loop Current, the simulation was to be restricted to seasons with clear thermal images. Altimeter tracking of the Loop Current was compared to the thermal images during the cooler seasons to determine its accuracy.

## Smoothing Techniques for Input Data

Input data from buoys, altimeter, and AVHRR are in the form of discrete data points. The simulation model required continuous input functions to drive the simulation. Therefore, continuous curve functions must be fit to the input data points. The team had to choose from several interpolation and forecasting techniques such as simple regression, Box - Jenkins techniques, and spline fits.

Second and third order regression fits were considered. These methods are easily implemented but cannot predict closed form curves such as eddy paths. The team also investigated elliptical curve fits. Elliptical fits have been fit to the Loop Current previously, and were well suited to the computer simulation model.

A combination of the autoregressive (AR) and moving average (MA) techniques were the Box - Jenkins method reviewed. This type of combination (ARMA) was already implemented in the existing computer simulation model to predict wave motion. Ideally, this existing algorithm would have been easily adapted to process the input data. [9,10]

Finally, a spline curve fit was considered. A spline method fits lower order curves to successive subsets of data points, which allows prediction of curves which do not have one to one correspondence between x and y coordinates. The Loop Current is such a curve. A spline fit required more processing of the data than other fitting techniques because some data points are processed more than once.

In the next chapter, the team will discuss the final design solution developed from the alternate approaches discussed in this chapter.

# PROGRAM MODEL AND METHODOLOGY

The team's goal was to develop a computer simulation model to predict the station keeping requirements of a semisubmersible platform in water deeper than 300 feet. Figure 2 shows an example platform. The team assumed that the platform is not anchored to the sea floor. Because the platform must remain relatively stationary for typical applications such as oil drilling, its engines, rather than anchors, must provide the station keeping forces necessary to hold the platform in place. This type of active station keeping is called dynamic positioning.

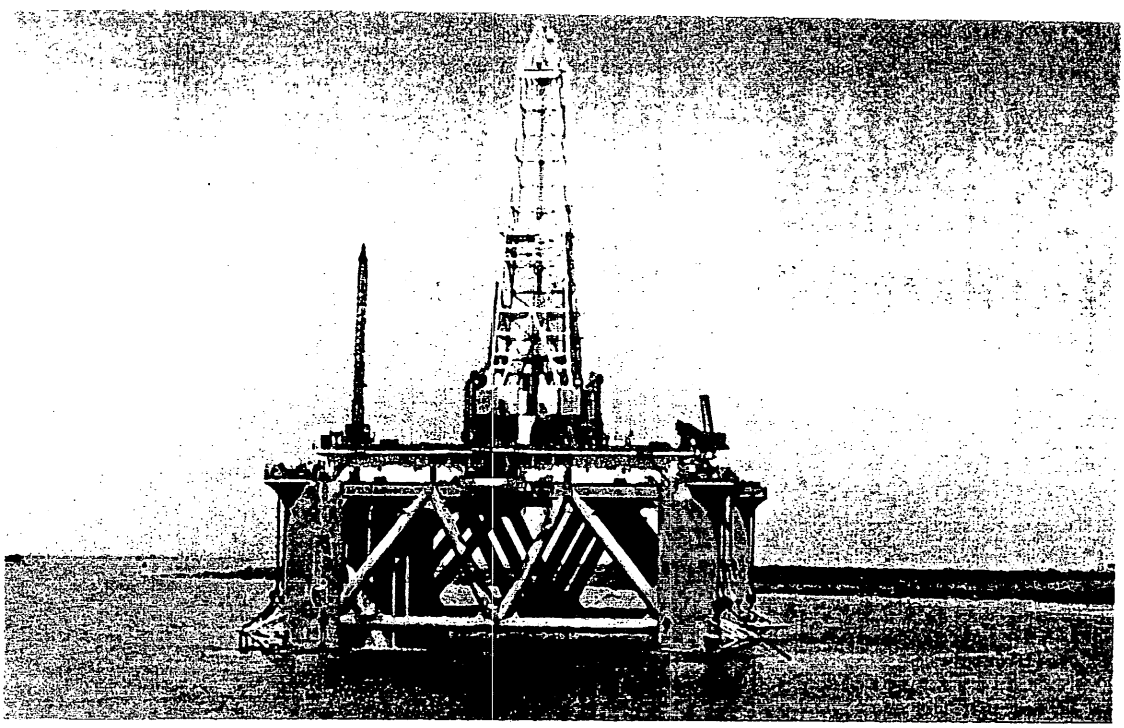


**FIGURE 2: AN EXAMPLE PLATFORM**

Several factors affect the platform's station keeping abilities. These factors are wind, waves, and currents. Wind causes an aerodynamic drag force on the superstructure of the platform. The wind drag force can cause significant drift in an unsecured platform. Waves cause the platform to translate vertically (heave). Waves can also have a directional effect, causing the platform to drift in the direction of the waves. Ocean currents create a hydrodynamic drag force on the hull of the platform. This hull force also causes the platform to move off station in the direction of the current.

All of these effects were simulated by the model in order to determine the requirements of the platform to maintain station. The requirements considered were engine power output and fuel supply. The simulated forces from the wind, waves, and currents were used in the model to calculate a net external force on the platform. This net external force was used to determine the power output required from the platform's engines in order to maintain station.

## Inputs to the Model

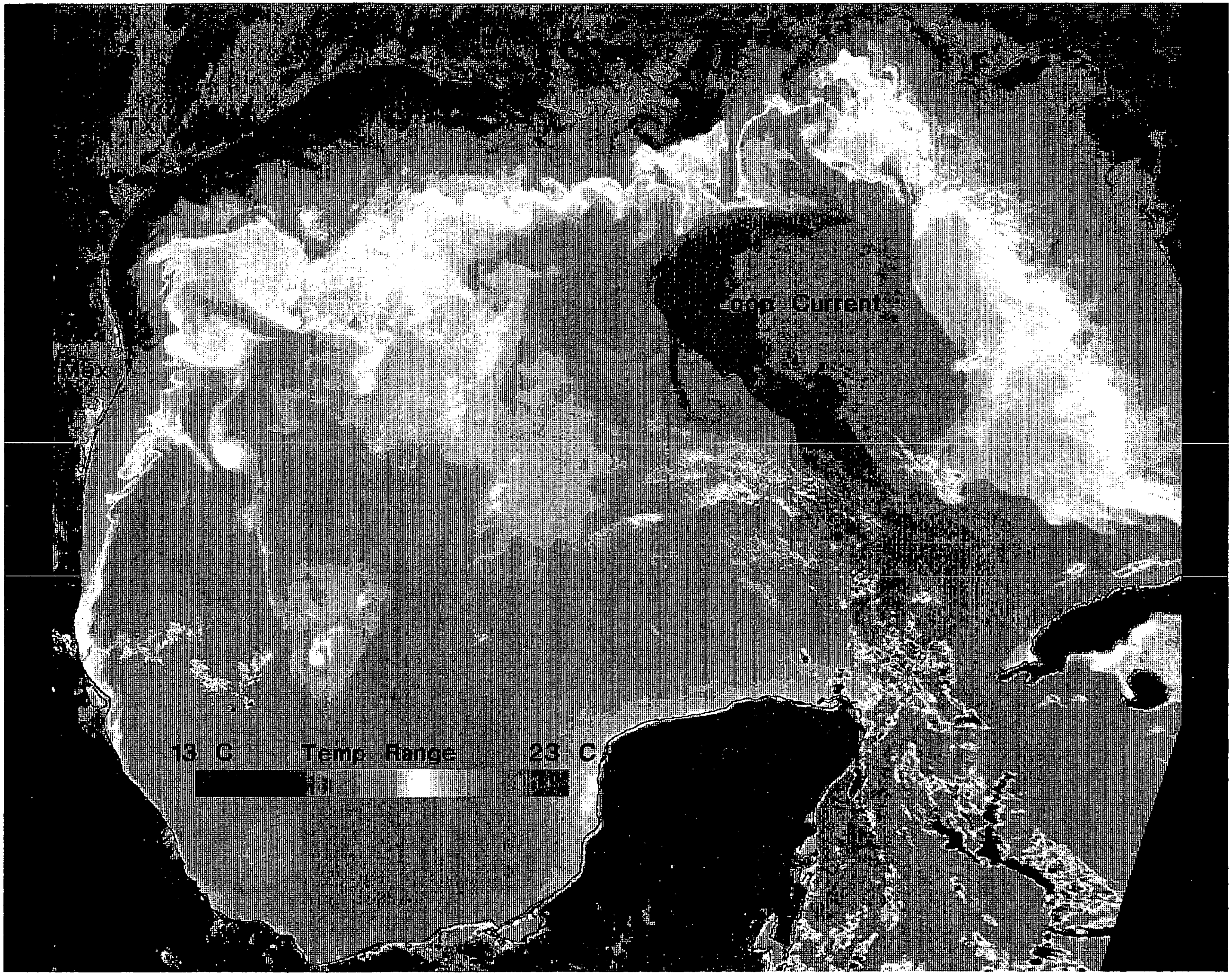
There were three areas of input to the model. The first area of input was magnitude and direction of ocean currents. The second area of input was weather conditions, which included wind speed and direction, wave height and wave period. Finally, the input parameters for simulation functions were platform response amplitude operators (RAO), ARMA model parameters, and event probability thresholds.

Ocean Currents. Magnitude and direction of ocean currents were determined from the advanced very high resolution radiometer (AVHRR) and buoy data. This radiometer is carried by the orbiting NOAA 9, 10 and 11 satellites. These satellites transmit AVHRR data to The University of Texas at Austin in the form of color images showing temperature distributions. Figure 3 shows a sample thermal image of the Gulf of Mexico. The AVHRR data was input for program developed to calculate velocities of currents. [6] This program models the shape of the current by drawing a series of connecting lines along the current's path. The points where the lines connect form distinct corners which mark specific locations in the current. As the current moves in successive images, the lines and corners are redrawn to reflect the new location. This program fits a vector from a corner point in one image to the next corner point in a successive image. This vector indicates the movement of a specific feature of the current. Figure 4 shows the linear image created by the program and the final vectors superimposed on a thermal image. By determining the distance traveled by a specific feature and the time to travel that distance, the velocities were calculated.

**FIGURE 3: SAMPLE THERMAL IMAGE OF THE GULF OF MEXICO**

**PICTURED ON NEXT PAGE (Page 18)**

This cover page has been used to avoid degrading the quality of the thermal image.



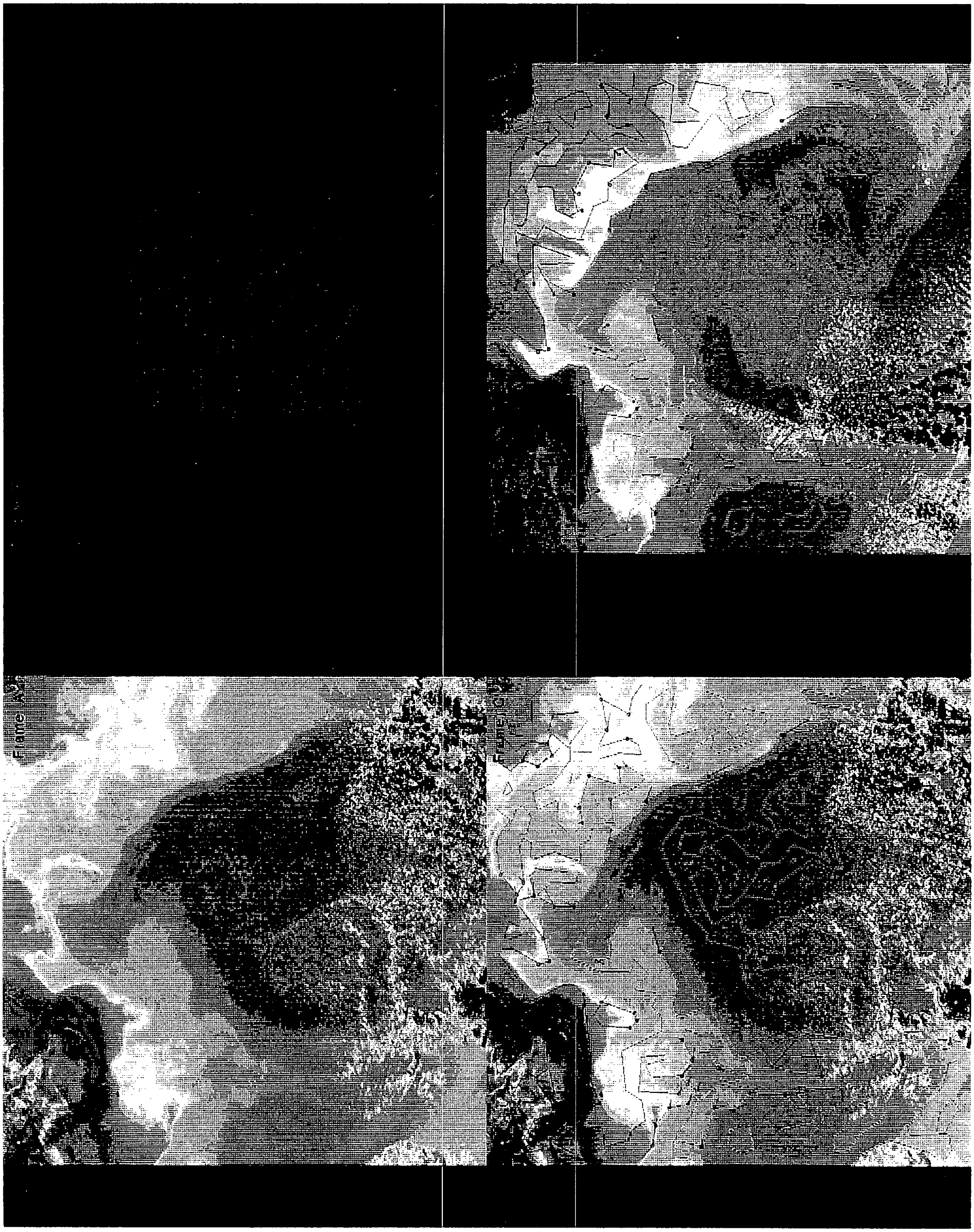

(Caption on previous page)

# FIGURE 4: FINAL VELOCITY VECTORS SUPERIMPOSED ON A THERMAL IMAGE OF THE GULF OF MEXICO

PICTURED ON NEXT PAGE (Page 20)

**Frame A: Initial Image**
**Frame B: Line Plot**
**Frame C: Velocity Vectors**
**Frame D: Successive Vectors**

This cover page has been used to avoid degrading the quality of the thermal image.



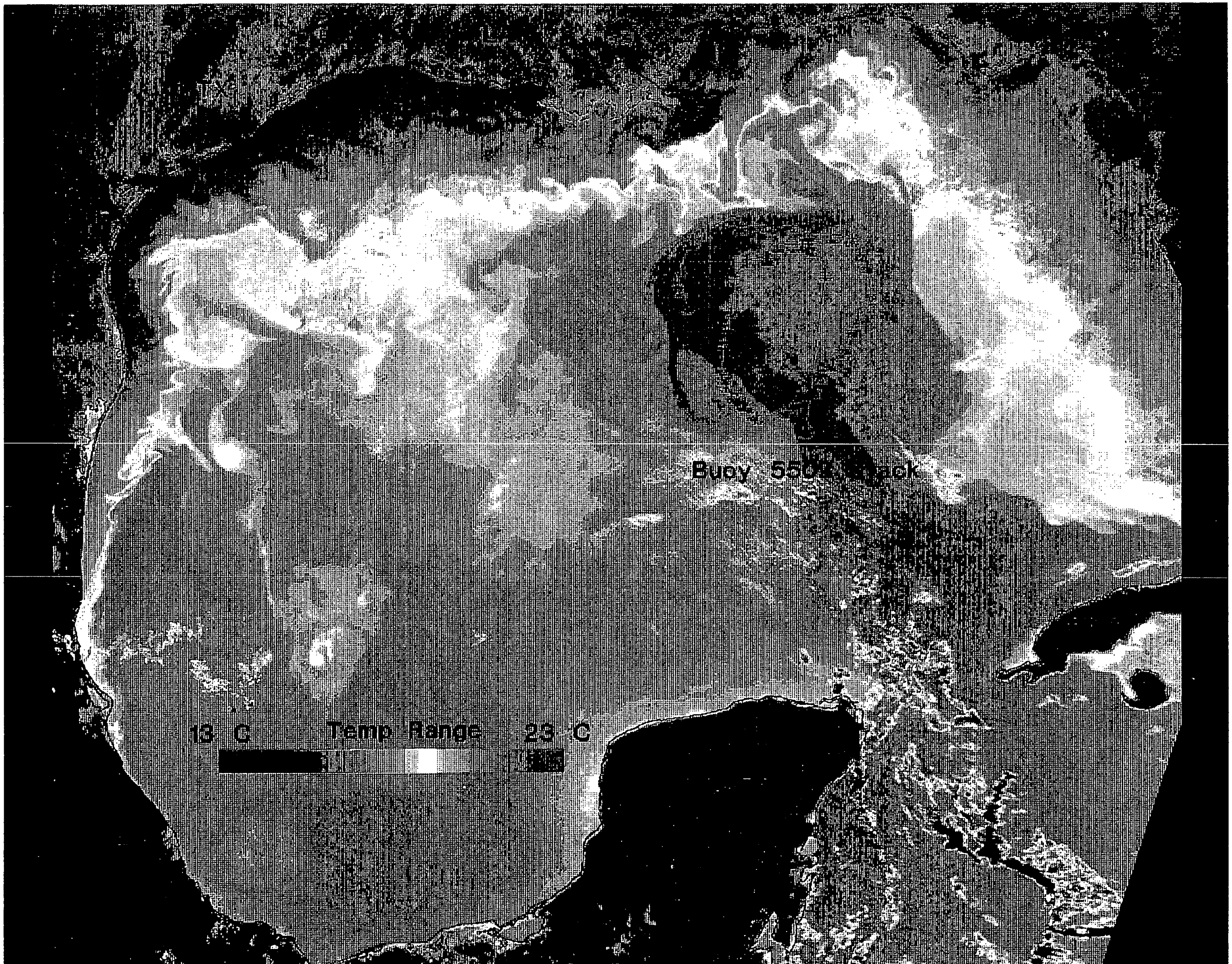
(Caption on previous page)

The buoy data was used in a manner similar to the thermal images to calculate the velocities of ocean currents in the Gulf of Mexico. Using a tracking program [11], a buoy trajectory was plotted to determine the movement of the current in which the buoy is moving. Two buoy trajectories were used to calculate the velocities. One buoy, labeled 3353, was located on the edge of the Loop Current in the eastern Gulf of Mexico. The other buoy, labeled 5502, was located in the middle of the Loop Current. Figure 5 shows the path of Buoy 5502 for 22 days. Several wild data points had to be edited from the buoy track. Points which could not be reasonably explained by motion of the current were considered wild points. Wild data points can be caused by such things as the satellite misreading the buoy location, the buoy drifting out of the current, and fishing boats catching the buoy in their nets. Once the buoy tracks were edited, the velocities of the currents were calculated by determining the distance the buoy traveled and dividing it by the time it took to travel that distance.

**FIGURE 5: THE PATH FOR BUOY 5502 FOR 22 DAYS**

**PICTURED ON NEXT PAGE (Page 23)**

This cover page has been used to avoid degrading the quality of the thermal image.




(Caption on previous page)

The velocities of the currents were calculated from the buoys using a linear approximation. Using this method for Buoy 3353, the team calculated the current velocity by taking the linear distance between two buoy locations and dividing it by the actual time between the two readings. This method does not account for the curvature of the eddy path. However, since Buoy 3353 had a reasonably linear path, a correction for the curvature of the path was not needed.

All data used to calculate the velocities of ocean currents was taken from the Spring of 1989. Specifically, the data for Buoy 3353 was taken from May to June. The AVHRR data used in the project was recorded from March 10 to March 13, 1989. Spring data was used because there is a greater temperature contrast between the Loop Current and the surrounding waters than during other seasons. The greater contrast allows clearer thermal images which show the Loop Current distinctly. The buoy data was available at UT/Austin beginning in February of 1989. This allowed buoy data to be taken from time periods corresponding to the thermal images.

The final results of the current velocity calculations were used to create an empirical probability distribution. The model used this distribution to predict the current velocity during the simulation time. The velocity values were magnitude only, and did not take into account location, season, or weather conditions.

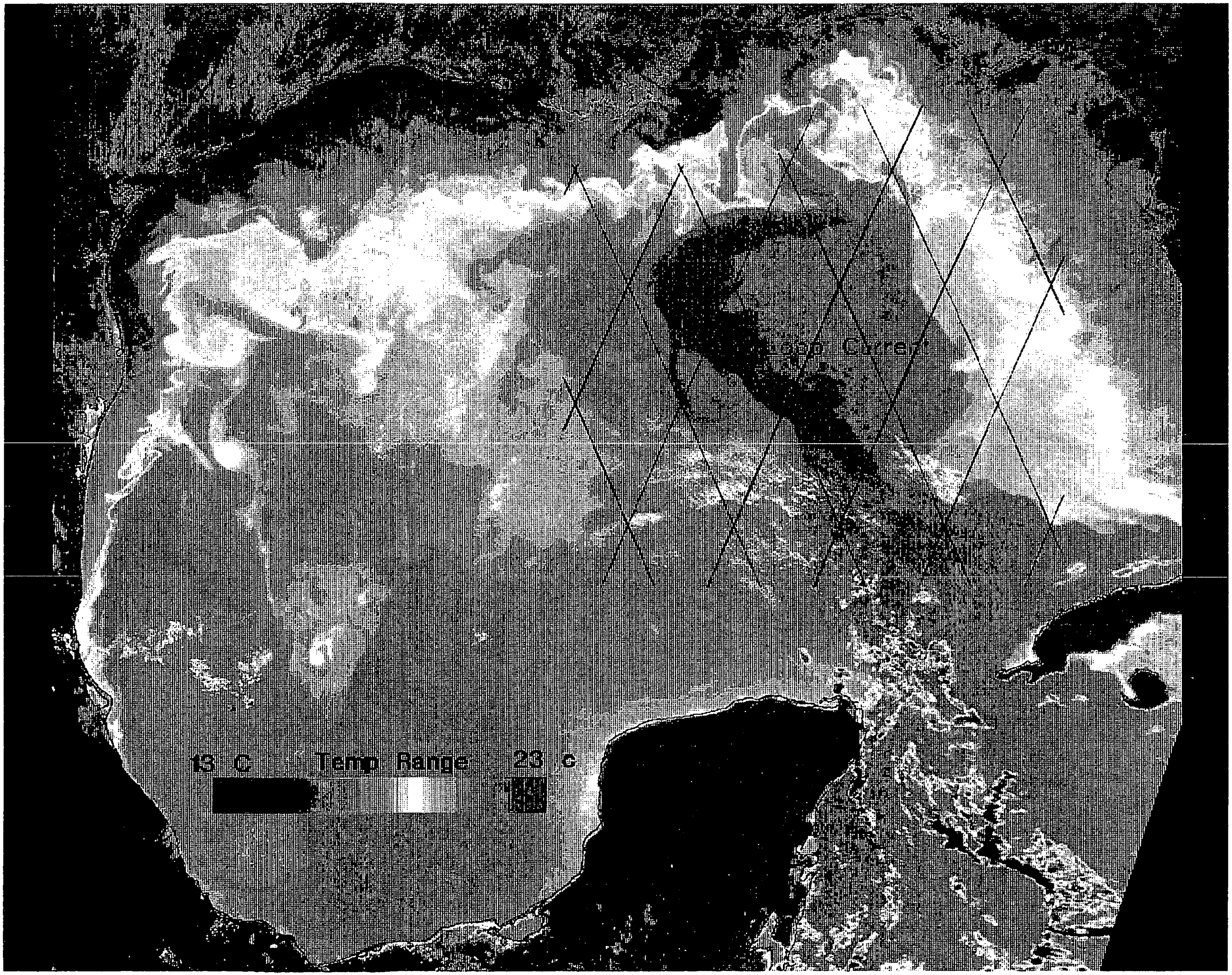
Weather. Weather information was used to predict wind and wave behavior in the Gulf of Mexico. The weather information came from the GEOSAT altimeter, NOAA AVHRR, Daily Weather Maps supplied by the Climate Analysis Center in Washington, D.C., and a composite weather model of weather conditions at Cape Hatteras, North Carolina. The team used these four sources together to model the weather related conditions for the simulation.

The GEOSAT satellite mission carried an altimeter which used a laser sensing ranger and locator to measure the surface conditions as the satellite passed over the the Gulf of Mexico. GEOSAT's altimeter was operational from January, 1986 to January, 1990. The data used in this project was of the first 150 days of 1989. The resolution of the GEOSAT altimeter was approximately one kilometer squared. GEOSAT took readings along a ground track approximately a kilometer wide. The same ground tracks were repeated every sixteen days. The team determined the location of the Loop Current using the AVHRR thermal images and defined a window to encompass the entire Loop Current. For the simulation model, the team used only the data from the tracks that were located in the window. Figure 6 shows the window and the altimeter tracks that fell within it.

## FIGURE 6: WINDOW AND THE ALTIMETER TRACKS THAT FELL WITHIN THE WINDOW

**PICTURED ON NEXT PAGE (Page 27)**

This cover page has been used to avoid degrading the quality of the thermal image.




(Caption on previous page)

The altimeter measured significant wave height at the ocean surface. From the significant wave height, the team was able to predict wind speed at the time of the reading using a factor provided in the altimeter data. The team averaged the readings along each individual track in the window and used these averages to estimate the wind and wave conditions throughout the entire window.

The Cape Hatteras composite weather model was made using data supplied by ARCO Oil and Gas Company from 1957. [7] The data was used to create time series models of wind speed and wave height, in addition to models for storm arrivals and durations.

The team used the Daily Weather Maps to chart the passage of weather fronts through the window. A figure of the Daily Weather Maps is included in Appendix H of this report. The days when fronts passed through the window were removed from the data set in order to leave only calm weather data. The data from these frontal passage days were placed in a separate file of storm data. It appeared that a better weather model could be built by separating the weather into calm periods and storms since these data were significantly different in magnitude.[2] The averaged values of wind speed and significant wave height for the calm tracks were plotted versus time and used to generate frequency histograms as shown in Appendix I. Neither the wind speed histogram nor the significant wave height histogram represented a distinct probability distribution clearly enough to warrant the fitting of a continuous probability function. Therefore, an empirical probability distribution was used to predict calm weather wind and wave behavior in the model.

The storm data suggested the storm magnitude and the time between storms. The team considered 1.5 meters the minimum average significant wave height for the sea conditions to be considered a storm. The storms were classed according to maximum wave height attained during the storm. Class 1 storms were those with wave heights up to two meters, and Class 2 storms were those with wave heights above two meters. According to the set of data points, storms occurred approximately twenty five percent of the time. The average interval between data points was about two and a half days. This suggested that storms arrived an average of every ten days. The team calculated the standard deviation of the time between storm arrivals as 8.76 days. The ten day interval was substantiated by the fact that fifteen storms occurred in the 150 available days of data.

Because of the limited number of data points available, no continuous probability functions were assigned to weather processes. Instead, the team used empirical probability distributions to predict the significant wave height and wind speed during both calm and storm conditions. Appendix I contains figures showing these cumulative probability distributions. These distributions were provided to the model in the form of one dimensional arrays.

Initial Conditions and Parameters. The weather and wave models in the simulation used Box - Jenkins time series analysis. These models were of the autoregressive moving average form. The autoregressive (AR) terms use previous values in the series, and the moving average terms (MA) use the current and previous random inputs. Together, these AR and MA terms are used to predict the next value in the time series. [7] The ARMA model is represented mathematically as

$z_t$ = theta(B)/phi(B) * $a_t$

where

$z_t$ = observation of the process at time t

mu = mean of the process

phi(B) = 1- $phi_1 B$ - $phi_2 B^2$ - ...-$phi_n B^n$ = autoregressive operator

theta(B) = 1 - $theta_1 B$ - $theta_2 B^2$ - ...-$theta_n B^n$ = moving average operator

B = backshift operator such that $Bz_t = z_{t-1}$

$a_t$ = random input or shock to the process at time t such that the mean of $a_t$ equals $sigma^2$ [7]

Wind speed and wave period during calm conditions were modeled with univariate ARMA models. These models are called univariate because the future time series values depend only on the previous values of that time series.

Wave height was modeled with a transfer function. Transfer function models represent the dependent variable as a function of an input variable and an ARMA noise model. The noise term is not necessarily the same as the corresponding univariate model of the dependent variable. The transfer function is represented mathematically as

$$z_t - mu = w(B)/delta(B) * \sim X_{t-b} - theta(B)/phi(B) * a_t$$

where

$z_t$ = observation of process at time t

mu = mean of process

$\sim X_{t-b} = X_{t-b} - mu_x$ = deviation of exogenous input variable about its mean at time t-b

$w(B) = w_0 - w_1 B - \ldots - w_n B^2$ = input lag operator of order n

$delta(B) = 1 - delta_1 B - \ldots - delta_r B^r$ = output lag operator of order r

theta(B) = moving average operator of order q

phi(B) = autoregressive operator of order p

B = backshift operator

$a_t$ = white noise random input [7]

Wave height is modeled using a transfer function. Wave height is the dependent variable and wind speed is the independent variable in the time series.

Intervention models are special cases of transfer models. Intervention models are used to model deterministic deviations from the mean of the process. The deterministic component models the change in the system as a step input. The dependent variable is a function of an intervention term and a random noise term. The intervention term takes a value of zero or one to determine whether the intervention variable is switched "on" or "off". [7] The mathematical form of the intervention model is represented below:

$$z_t - mu = w(B)/delta(B) * I_{t-b} - theta(B)/phi(B) * a_t$$

where

$z_t$ = observation of process at time t

mu = mean of process

$I_{t-b}$ = zero-one variable denoting whether an impact or intervention variable is switched "on" or "off" at time t-b

w(B) = input lag operator of order s

delta(B) = output lag operator of order r

theta(B) = moving average operator of order q

phi(B) = autoregressive operator of order p

B = backshift operator

$a_t$ = white noise random input [7]

Storms are modeled in the simulation with an intervention model. The intervention term is set to zero during calm weather and is set to one during storm conditions. [2]

The simulation model also required the response amplitude operators for the simulated vessel. RAO's are generally presented as curves plotted versus wave period. Figure 7 shows an example RAO plot. The model read the RAO's as a series of linear approximations over short intervals of wave period values. The RAO's were required to predict the vessel's heave response to waves. The simulated vessel was a Western Pacesetter #2 semisubmersible platform manufactured by Friede and Goldman in New Orleans, Louisiana. ARCO Oil and Gas Company provided the team with the RAO's for the Pacesetter.

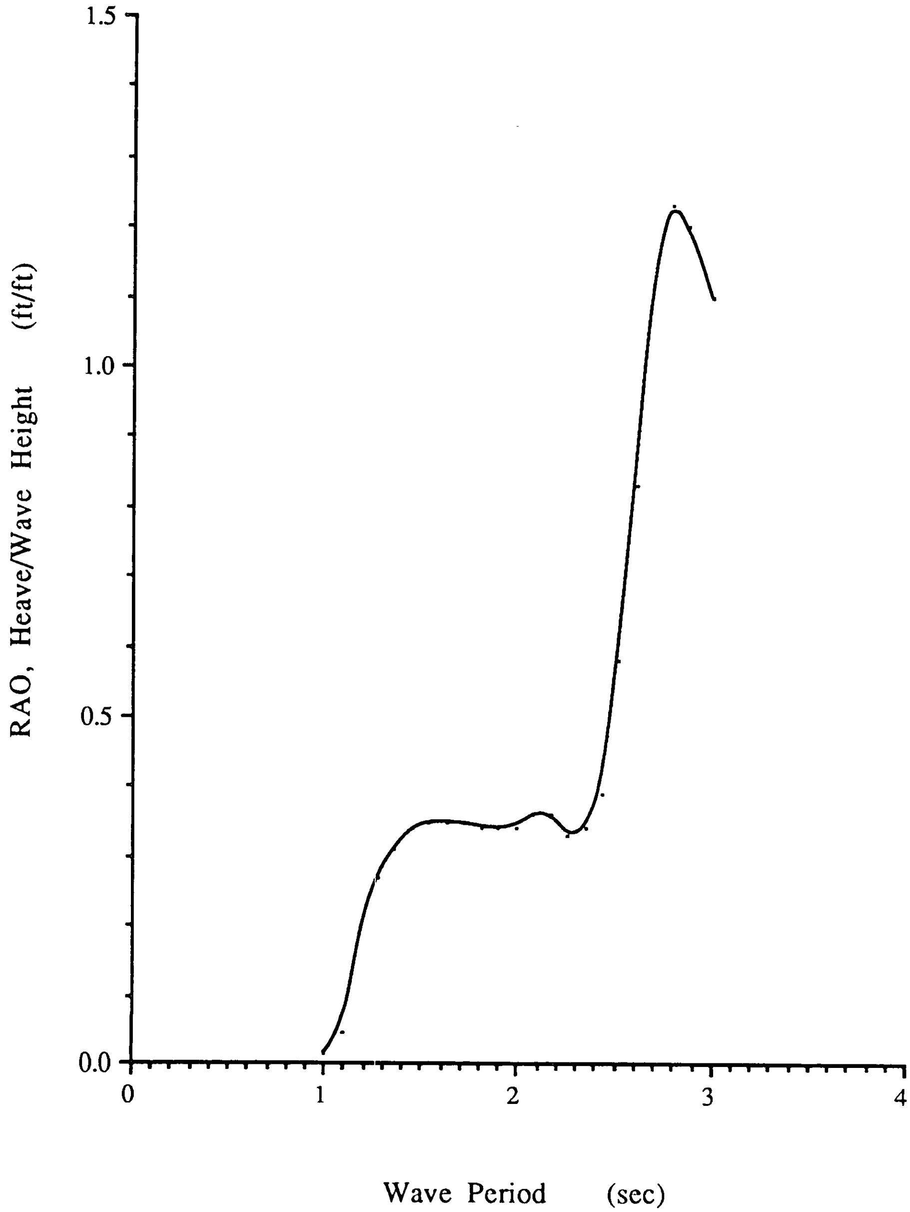



**FIGURE 7: EXAMPLE RAO PLOT**

5

# PROGRAM DEVELOPMENT

The team's simulation model was developed from an existing model. The existing model simulated the operation of an oil drilling vessel in the Arctic Ocean. The original model was written in a combination of SLAM and FORTRAN programming languages for use on the UT/Austin CDC Cyber system . Several changes were made to the original model. These changes are as follows:

1. Code written in SLAM was deleted from the model, leaving only FORTRAN code.
2. Oil drilling operations were deleted from the model.
3. Wind and current drag force models were written for the model.
4. A time keeping routine was developed to replace the deleted SLAM time keeping functions.
5. The weather and wave input values were changed to fit conditions in the Gulf of Mexico.
6. Routines were developed to generate random numbers from uniform and normal probability distributions.

SLAM was deleted for three reasons. First, SLAM is not as widely available as FORTRAN. Thus, by using only FORTRAN in the simulation, the team felt that the program would be more applicable to a wider range of users. Second, SLAM requires a large amount of computer memory (600 sectors on the CDC Cyber system). The team hoped to adapt the program for possible use on a personal computer, where computer memory is more limited than on a mainframe computer. Finally, by writing the program in one language instead of two, the team made the program easier to compile and run, since no linking was involved.

The original simulation modeled a platform engaged in offshore oil drilling operations. Since the project originally was intended to apply to non drilling operations such as the U.S. Navy's DORI project, the team felt that drilling operations should be deleted. Also, drilling operations comprised the majority of the original simulation. Therefore, deleting the drilling allowed the program to run in a much shorter amount of time in addition to increasing the program's range of applications.

The team created functions which calculated the force on the platform due to both wind drag and current drag. These functions were based on tests of the performance of the Western Pacesetter. [12] The platform was assumed to maintain station in the original model, therefore the original model did not include any drag force calculations. The new functions determine the wind and current conditions given the specific location of the platform. These conditions are then used to calculate the net drag force on the platform. From the net drag on the platform, the engine power output requirement was calculated as a percentage of maximum available power (6000 horsepower).

Figure 8 shows as simple flow chart of the inputs to the model.

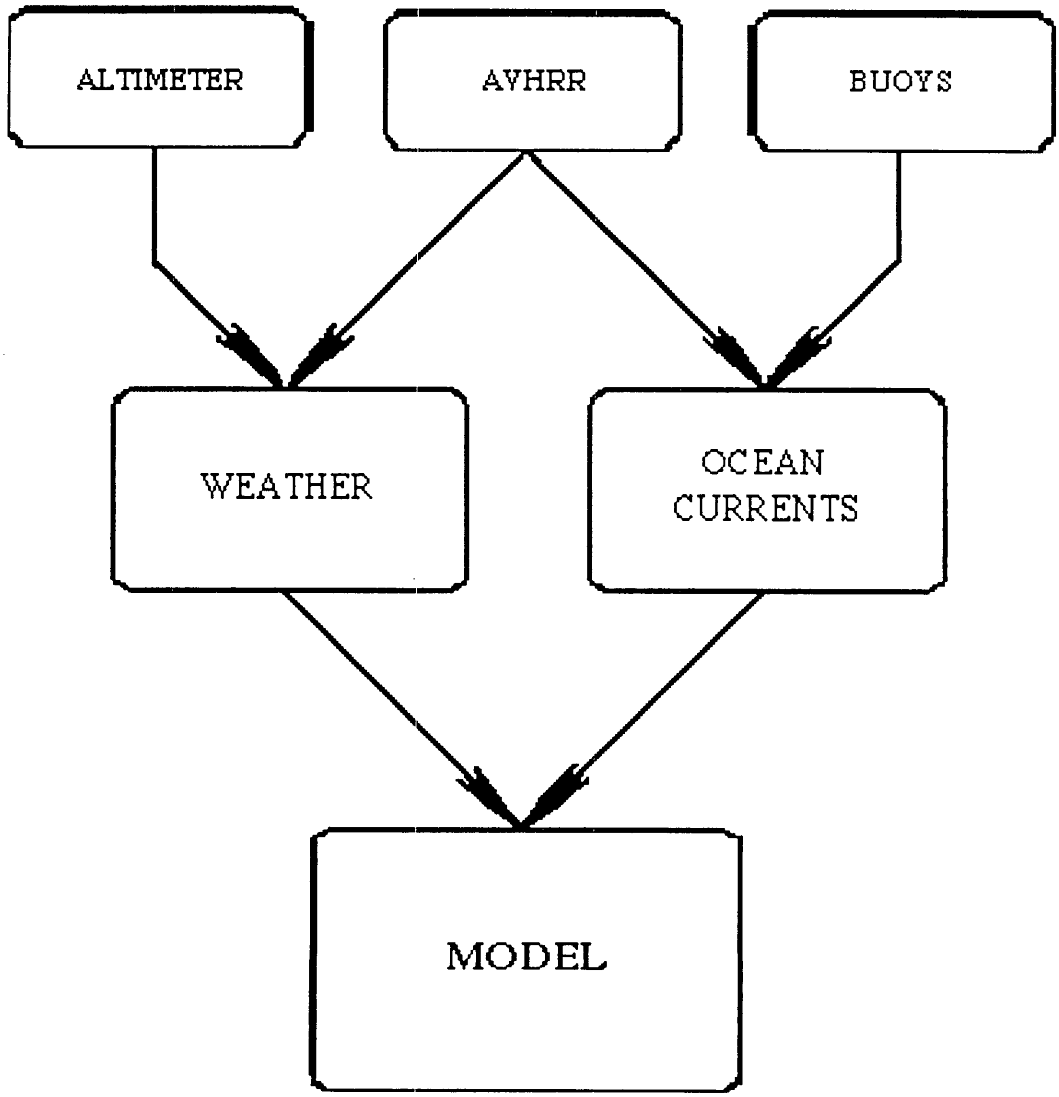



FIGURE 8: SIMPLE FLOW CHART
OF INPUTS

Without the SLAM time keeping routines, the computer simulation model only goes through one iteration. Therefore, the team developed a new time keeping routine to allow the program to simulate periods of time longer than one interval. This interval best fit the input data available to the team. The time keeping routine was a simple loop structure which updates the state of the platform and external conditions at each time interval. The team wrote the simulation with the time interval as a variable so that various intervals could be used. Figure 9 shows a simple flow chart of the structure of the program.

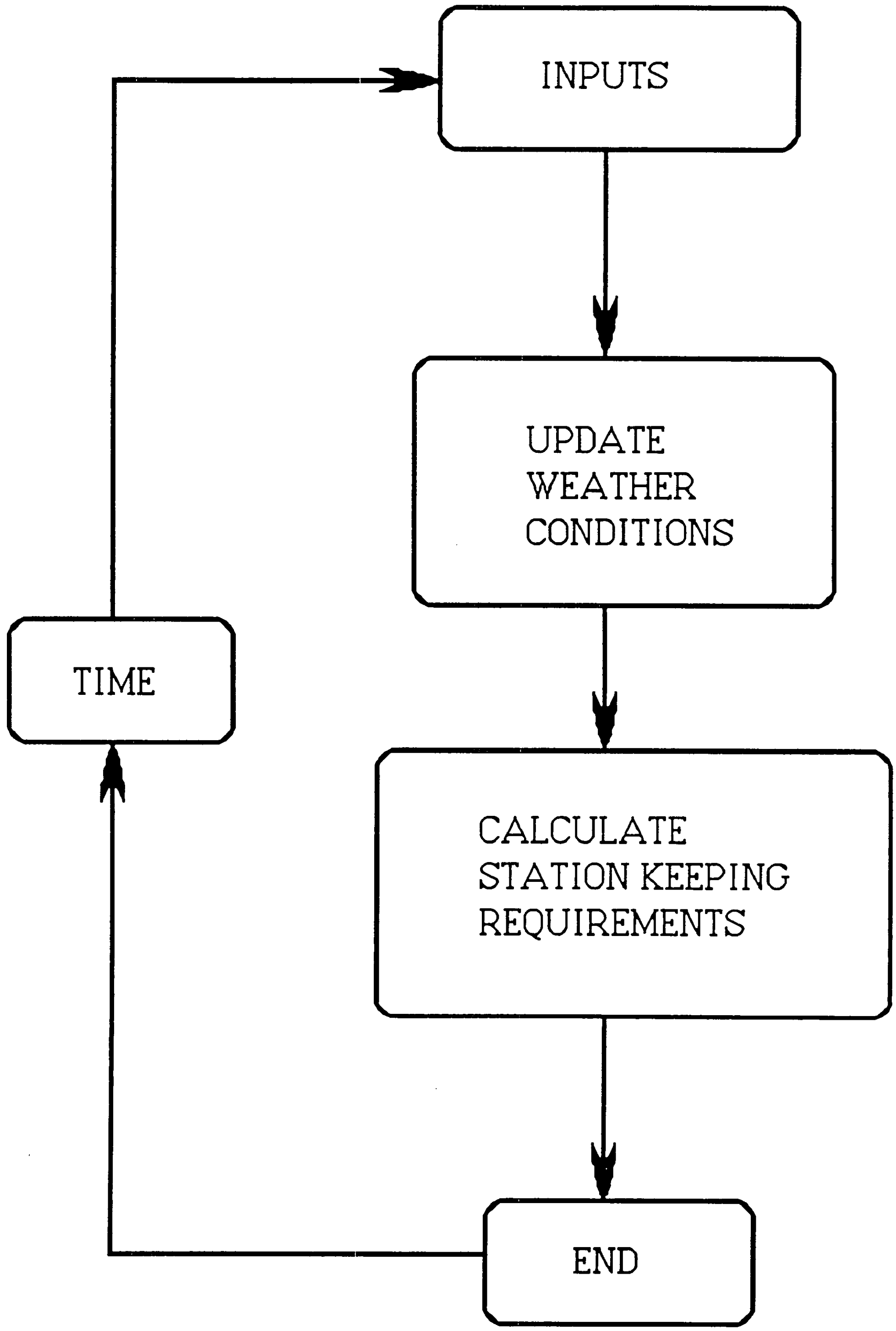



**FIGURE 9: SIMPLE FLOW CHART OF THE STRUCTURE OF THE PROGRAM**

Input parameters for the weather and wave models were originally developed for the Arctic Ocean. Since the team simulated operations in the Gulf of Mexico, these parameters had to be altered. Wind conditions, wave conditions, and storms in the Gulf of Mexico are all quite different than their Arctic counterparts. For instance, storms (excluding hurricanes) are generally less severe in the Gulf of Mexico than in the Arctic Ocean. Therefore, the team only included two classes of storms, rather than the original three.

The simulation model required random number inputs for the weather and wave condition models. Therefore, the team wrote functions which generated random numbers according to uniform and normal probability distributions. In a uniform distribution, each number in the range has an equal probability of occurring. The uniform random number function used the Fibonacci sequence to generate pseudorandom numbers between zero and one. Five significant digits were used to give a cycle repeat length of at least 150,000 terms. The first 1000 terms were deleted, and every second term in the series was used to increase the apparent randomness of the numbers. The normal probability distribution function is commonly graphed as a "bell curve". The normal function generates a normally distributed random number with a mean of zero and a variance of one by adding twelve numbers from the uniform function and subtracting six from the sum.

A flowchart of the complete model is included in the Appendix section of this paper with listing of the complete computer program.

# RESULTS

Based on the input data given, the model predicts weather, wave, and current conditions and the vessel power output required to maintain station.

The weather, wave, and current predictions that the model generates are reasonable estimates of conditions that could be expected in the Gulf of Mexico. The team compared the values generated by the simulation to the actual input data points, and found that these output values did fall in the range of the actual readings. Therefore, the team feels the simulation's models for wind, wave, and current are reasonable. The simulated values are general approximations of the conditions across the window, however, not accurate point values.

The simulation output for power required to maintain station was modeled assuming a worst case scenario. In this scenario, the team assumed that all forces on the platform acted concurrently and in the same direction. This scenario yields the greatest net force on the platform for a given set of environmental conditions. Since no actual semisubmersibles are operating unanchored at this time, the team had no real data with which to compare the simulation results. Therefore, the accuracy of this model is unknown. However, since the environmental conditions were assumed to be worst case and conservative engine power estimates were used, the team believes that any error in simulation results will tend to be conservative. In other words, the predicted station keeping requirements will most likely be overestimated, meaning that a platform should be able to maintain station more easily than the simulation predicts.

# CONCLUSIONS

This computer simulation model was developed to simulate the operation of a semisubmersible platform in deep waters of the Gulf of Mexico. This computer simulation model will be valuable to any organization interested in keeping a dynamically positioned vessel stationary in the Gulf of Mexico. The simulation is versatile and simple to use. Because the simulation does not contain any drilling operations, its uses are not limited to the oil industry. Also, the model can simulate any type of dynamically positioned vessel if the vessel's RAO's are known. By varying the weather input, the simulation can model all four seasons. This allows for modeling of a complete year, or whichever portion of the year is needed.

The simulation is easy to run. It requires only three input files: one for ARMA model parameters, one for weather and wave probability thresholds, and one for initial values for the time series functions. Therefore, to change the conditions of the simulation, only the input files need to be modified, rather than the program code. However, if the code does need to be altered or expanded, the FORTRAN language used is widely known and the code is well documented.

## RECOMMENDATIONS

The main problem that the team encountered was limited data sets available within the time available to this project. The team was only provided with 150 days of altimeter data from the GEOSAT satellite mission. A more complete data set would provide more accurate probability distributions for wind, waves, and storms. Ideally, data from several years could be used to generate input for the model. This would allow the simulation to differentiate between seasons.

The team did not have access to the complete set of GEOSAT data. At the time of the project, the orbital errors inherent in this data had not yet been corrected. Therefore, the team was unable to make use of the macroscopic sea height readings, which show the overall height of the sea surface. Since the surface of warm water features like the Loop Current is higher than the surrounding cooler waters, the height readings could be used to locate the Loop Current and warm core eddies. The altimeter location of these features would be useful in the summer months when the thermal images are indistinct. The team recommends that future research be devoted to this use of the altimeter data.

AVHRR and buoy data was only available at UT/Austin for the past year, starting on February 28, 1989. Three days of AVHRR data were used for this project to compute velocities of currents. A larger data set could be used to create a finer grid structure for the current array which would provide a more precise estimate of the current at any specific location. Due to the amount of time required to track and edit the buoy paths, a limited number of paths were processed for this project. More buoy tracks could substantiate the elliptical path patterns observed during this project as well as provide more data on velocities of currents. The team recommends that in future projects the elliptical fitting program available from Glenn and Forrestall be used to evaluate the buoy paths. The program was not available in time to use for this project.

The wind, wave, and current models include only magnitudes. The team did not have time to model the directionality of these phenomena. Directionality can be modeled using sectors to denote direction from a specific location. The size of the sectors is arbitrary, and the sectors need not be equal in size. The team suggests using a Markov Chain model to model the probabilities of future directions based on the previous directions. Some of the direction variables may need to be synchronized because of their effect on one another. For example, wind and wave directions are not always the same, but are often related to one another. This approach has been successful in modeling directionality of wind and waves in Alaska. [7] Since directionality has such a great effect on the station keeping of the platform, the team recommends that further research be conducted in this area.

The team did not model the movement of the platform caused by environmental forces, except to tell the user whether the platform is able to keep station. The team recommends that future models include drift of the platform by modeling the platform's location on a grid map. Such a location model would allow a much more complete treatment of current velocity magnitudes and directions, since these values vary with geographic location. By modeling location, future researchers could also incorporate station keeping strategies. These strategies could vary with the amount of drift allowed for the platform. For example, allowing larger amounts of drift would make possible a sprint and drift strategy, where the platform "sprints" into the current to the edge of the allowed drift area and then "drifts" with the current to the opposing edge.

Future users of the model are likely to be interested in drilling for oil. In order to make the model more useful to the oil industry, drilling operations could be included in the simulation. Drilling operations could be included in the model either by reinstating the SLAM and FORTRAN drilling routines or by writing new FORTRAN code to simulate drilling.

The team developed the model for use on UT/Austin's CDC Cyber system, which will be removed from operation in January 1991. In order to avoid losing the simulation, it must be moved to another system. The team recommends that further study be devoted to the translation of the program to a version of FORTRAN compatible with other systems. Both mainframe systems such as UT/Austin's VAX and personal computers such as IBM PC's should be considered.

**REFERENCES**

1. NASA/USRA University Advanced Design Program. Handbook for Students, 1988-89.
2. Hoffman, Susan Elizabeth. An Integrated Model of Drilling Vessel Operations. Master's Thesis, University of Texas at Austin, December,1982.
3. Crawford, Dr. Melba M. Conversations throughout the semester. Department of Mechanical Engineering, University of Texas at Austin, Spring 1990.
4. Lunderg, Dr. John. Conversations throughout the semester. Department of Aerospace Engineering, University of Texas at Austin, Spring 1990.
5. Moran, David. Conversation on March 22, 1990. David Taylor Labs, Bethesda, Maryland.
6. O'Keefe, Shawn. An Automated Method for Estimating Oceanic Flow Fields from Satellite Imagery. Doctoral Dissertation, University of Texas at Austin, December, 1989.
7. Crawford, Dr. Melba M. Time Series Models of Sea State Variables. Department of Mechanical Engineering, University of Texas at Austin, March, 1985.
8. Daily Weather Maps - February 27 through June 4, 1989. Climate Analysis Center, Washington, D.C. Obtained through Sonia Gallegos, Ph.D.
9. Hoff, John C., PhD. A Practical Guide to Box-Jenkins Forecasting. Lifetime Learning Publications, Belmont, California, 1983.

10. Bowerman, Bruce L. and Richard T. O'Connell. Forecasting and Time Series. Duxbury Press, North Scituate, Massachusetts, 1979.

11. Tracks. Program available at University of Texas Center for Space Research Imaging Lab. Department of Mechanical Engineering, University of Texas at Austin.

12. Greiner, W. and D. Tuturea. Preliminary Operability Analysis for Western Pacesetter III Conversion. Brown & Root U.S.A., Houston, October 25, 1989.

# APPENDIX A

# EQUATION DEVELOPMENT

# Velocity Calculations for Currents

1. convert latitudes and longitudes into minutes
   $= \text{Degree} \times 60 \frac{\text{min}}{\text{deg}} + \text{minutes}$
2. calculate the change in latitude and longitude between two points (in minutes)
   $\Delta \text{Lat} = \text{Lat}1 - \text{Lat}2$
   $\Delta \text{Long} = (\text{Long}1 - \text{Long}2) * \cos(\Delta \text{Lat})$
3. calculate the distance between two points (in miles)
   $1 \text{ min} = 1 \text{ mile}$
   $\text{distance} = (\Delta \text{Lat}^2 + \Delta \text{Long}^2)^{1/2}$
4. calculate the velocity magnitude (miles per hour)
   $\text{vel} = \text{distance} / \Delta \text{time}$
   $\Delta \text{time}$ = time to travel between two points
5. calculate velocity directions
   $\theta = \tan^{-1} \frac{\Delta \text{Lat}}{\Delta \text{Long}}$
   $\text{vel}\,x = \pm \text{vel} * \sin\theta$ (or $\cos\theta$)
   $\text{vel}\,y = \pm \text{vel} * \cos\theta$ (or $\sin\theta$)

# APPENDIX B

# SAMPLE OUTPUTS

B1

```
                            INPUT ECHO REPORT

       INPUT FOR ARMA MODELS
 MODEL    P       Q    DELTA SIGMA NSAMP
   1      1       1    9.004 1.000   300
PHI S=  .189
THETA S= -.277
 MODEL    P       Q    DELTA SIGMA NSAMP
   2      1       0    3.662 1.000   300
PHI S=  .466

INTERVENTION WEIGHTS
    HEIGHT      PERIOD
    3.5888       .1876
    6.1444      1.4865
    6.9556      2.1046

TRANSFER FUNCTION MODEL INPUT
NOISE P  NOISE Q  OUTPUT ORDER   INPUT ORDER   INPUT BACKSHIFT   SIGMA  MEAN
  1        0            2              1              0          1.000 7.262
PHI S=  .418
DELTA S= 1.145 -.193
OMEGA S=  .162  .139

NO. OF TIME BETWEEN STORMS  STORM LENGTHS  FOR MONTH
                  2                 15           1
                 19                 14           2
                 16                  7           3
                 15                 10           4
                 14                  7           5
                  3                  2           6
                  3                  3           7
                  9                  5           8
                 17                 12           9
                 25                 16          10
                 23                 14          11
                 18                 13          12

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  1
  .04   .07   .15   .22   .30   .37   .41   .44   .48   .52
  12.   14.   20.   22.   24.   26.   28.   34.   36.   40.
  .59   .63   .67   .70   .78   .81   .85   .93   .96  1.00
  44.   50.   58.   60.   62.   84.   88.  106.  108.  128.
    0     0     0     0     0
    0     0     0     0     0

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  2
  .04   .08   .17   .21   .29   .33   .38   .46   .50   .58
   6.   14.   16.   24.   30.   44.   48.   60.   70.   74.
  .63   .67   .71   .79   .83   .88   .92   .96  1.00     0
  80.   82.   84.   88.   94.  100.  110.  124.  222.     0
    0     0     0     0     0
    0     0     0     0     0
```

ORIGINAL PAGE IS
OF POOR QUALITY

B2

```
CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  3
   .05    .09    .14    .27    .36    .41    .45    .55    .64    .73
   20.    22.    26.    30.    32.    34.    38.    40.    44.    56.
   .77    .82    .86    .91    .95   1.00      0      0      0      0
   66.    70.    92.   106.   124.   126.      0      0      0      0
     0      0      0      0      0
     0      0      0      0      0

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  4
   .06    .13    .19    .25    .38    .44    .50    .56    .63    .69
   12.    22.    24.    26.    30.    32.    42.    50.    56.    68.
   .75    .81    .88    .94   1.00      0      0      0      0      0
   74.    76.   116.   184.   202.      0      0      0      0      0
     0      0      0      0      0
     0      0      0      0      0

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  5
   .07    .13    .20    .33    .40    .47    .53    .60    .67    .73
   10.    14.    16.    20.    22.    26.    54.    94.   120.   132.
   .80    .87    .93   1.00      0      0      0      0      0      0
  138.   146.   148.   158.      0      0      0      0      0      0
     0      0      0      0      0
     0      0      0      0      0

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  6
   .33    .67   1.00      0      0      0      0      0      0      0
   16.    70.    74.      0      0      0      0      0      0      0
     0      0      0      0      0      0      0      0      0      0
     0      0      0      0      0      0      0      0      0      0
     0      0      0      0      0
     0      0      0      0      0

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  7
   .33    .67   1.00      0      0      0      0      0      0      0
   54.    86.   328.      0      0      0      0      0      0      0
     0      0      0      0      0      0      0      0      0      0
     0      0      0      0      0      0      0      0      0      0
     0      0      0      0      0
     0      0      0      0      0

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  8
   .11    .22    .33    .44    .56    .67    .78    .89   1.00      0
   14.    36.    52.    90.   132.   152.   218.   292.   300.      0
     0      0      0      0      0      0      0      0      0      0
     0      0      0      0      0      0      0      0      0      0
     0      0      0      0      0
     0      0      0      0      0

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH  9
   .05    .09    .14    .18    .23    .27    .45    .55    .59    .64
   12.    18.    24.    28.    32.    36.    40.    42.    56.    58.
   .68    .73    .77    .82    .91    .95   1.00      0      0      0
   80.    82.   168.   188.   296.   374.   578.      0      0      0
     0      0      0      0      0
     0      0      0      0      0
```

ORIGINAL PAGE IS
OF POOR QUALITY

B3

```
CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH 10
  .02   .10   .17   .24   .29   .40   .48   .50   .52   .55
   8.   12.   16.   18.   20.   22.   24.   26.   28.   32.
  .60   .62   .64   .67   .71   .74   .76   .83   .86   .88
  36.   38.   40.   44.   50.   54.   56.   60.   62.   66.
  .90   .93   .95   .98  1.00
  76.   86.  100.  108.  116.

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH 11
  .04   .07   .11   .18   .21   .25   .29   .36   .43   .46
  10.   16.   18.   20.   24.   26.   28.   32.   36.   38.
  .50   .54   .61   .64   .68   .71   .75   .79   .82   .89
  40.   46.   48.   52.   54.   58.   64.   70.   72.   80.
  .93   .96  1.00     0     0
  82.   86.   88.     0     0

CDF CUTOFFS & TIME BETWEEN STORMS FOR MONTH 12
  .03   .06   .12   .15   .27   .39   .45   .55   .61   .64
  14.   16.   22.   24.   26.   28.   30.   38.   42.   46.
  .70   .79   .82   .85   .91   .94   .97  1.00     0     0
  48.   52.   54.   56.   64.   66.   74.   90.     0     0
            0     0     0     0
            0     0     0     0

CDF CUTOFFS & STORM LENGTHS FOR MONTH  1
 .04 .11 .19 .22 .41 .56 .63 .70 .74 .81 .85 .89 .93 .961.00    0
  4.  6.  8. 10. 14. 18. 20. 22. 26. 34. 38. 56. 58. 62. 66.    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH  2
 .08 .25 .42 .46 .63 .67 .71 .75 .79 .83 .88 .92 .961.00    0    0
  4.  6.  8. 10. 12. 14. 16. 18. 20. 22. 24. 28. 64. 66.    0    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH  3
 .27 .41 .68 .73 .77 .911.00   0   0   0   0   0   0   0   0    0
  6.  8. 10. 12. 14. 20. 40.   0   0   0   0   0   0   0   0    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH  4
 .06 .19 .50 .56 .69 .75 .81 .88 .941.00   0   0   0   0   0    0
  4.  6. 10. 12. 18. 20. 30. 34. 40. 48.   0   0   0   0   0    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH  5
 .07 .13 .40 .67 .87 .931.00   0   0   0   0   0   0   0   0    0
  4.  6.  8. 10. 12. 16. 20.   0   0   0   0   0   0   0   0    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH  6
 .331.00   0   0   0   0   0   0   0   0   0   0   0   0   0    0
  8. 10.   0   0   0   0   0   0   0   0   0   0   0   0   0    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH  7
 .33 .671.00   0   0   0   0   0   0   0   0   0   0   0   0    0
  4.  8. 10.   0   0   0   0   0   0   0   0   0   0   0   0    0
```

B4

```
CDF CUTOFFS & STORM LENGTHS FOR MONTH  8
 .11 .22 .67 .781.00   0    0    0    0    0    0    0    0    0    0    0
  4.   6.  8. 10. 22.  0    0    0    0    0    0    0    0    0    0    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH  9
 .05 .23 .36 .50 .64 .68 .73 .77 .82 .91 .951.00    0    0    0    0
  4.   6.  8. 10. 12. 14. 18. 20. 22. 24. 30. 32.    0    0    0    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH 10
 .02 .21 .38 .48 .57 .69 .74 .76 .79 .81 .86 .88 .93 .95 .981.00
  4.   6.  8. 10. 12. 14. 16. 20. 22. 28. 30. 34. 36. 40. 58. 72.

CDF CUTOFFS & STORM LENGTHS FOR MONTH 11
 .04 .07 .18 .32 .43 .50 .54 .68 .71 .86 .89 .93 .961.00    0    0
  2.   6.  8. 10. 12. 14. 16. 20. 22. 24. 30. 36. 38. 40.    0    0

CDF CUTOFFS & STORM LENGTHS FOR MONTH 12
   0 .09 .21 .30 .42 .61 .70 .82 .85 .88 .91 .94 .971.00    0    0
  4.   8. 10. 12. 14. 16. 18. 22. 30. 52. 58. 74. 76.    0    0    0

CDF CUTOFFS FOR STORM CLASS

MONTH  1     2
  1    .15   .55
  2    .29   .96
  3    .32   .68
  4    .38   .50
  5    .53   .93
  6    .67  1.00
  7   1.00     0
  8    .44   .78
  9    .27   .73
 10    .21   .64
 11      0   .46
 12    .18   .61

STARTING MONTH IS  8 AND DAY IS 15

MON DAY WINDSPEED  WAVEHT  WAVEPD HEAVE  PROPUL  DRIFT
 8   15 11.884174   9.743  8.120  2.30   .6679    I

MON DAY WINDSPEED  WAVEHT  WAVEPD HEAVE  PROPUL  DRIFT
 8   15 11.149488 11.001   5.996  1.12   .6585    I

MON DAY WINDSPEED  WAVEHT  WAVEPD HEAVE  PROPUL  DRIFT
 8   15 10.130361   9.762  6.718  1.53   .6463    I

MON DAY WINDSPEED  WAVEHT  WAVEPD HEAVE  PROPUL  DRIFT
 8   15 11.657371   8.997  6.397  1.20   .6649    I

MON DAY WINDSPEED  WAVEHT  WAVEPD HEAVE  PROPUL  DRIFT
 8   15 13.784485 10.105   8.556  2.57   .6952    I

MON DAY WINDSPEED  WAVEHT  WAVEPD HEAVE  PROPUL  DRIFT
 8   15 12.916946   9.829  6.887  1.65   .6823    I
```

B5

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 10.618981 | 9.618 | 7.916 | 2.18 | .6520 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 9.393722 | 8.515 | 7.320 | 1.66 | .6383 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 10.849476 | 8.584 | 7.596 | 1.81 | .6548 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 10.652271 | 8.107 | 5.842 | .73 | .6524 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 10.93[illegible]468 | 7.658 | 4.970 | .24 | .5891 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 11.425301 | 7.715 | 6.018 | .80 | .5953 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 10.119716 | 7.612 | 5.881 | .71 | .5795 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 11.347090 | 6.315 | 5.964 | .63 | .5943 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 10.481868 | 7.400 | 6.781 | 1.19 | .5837 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 11.858495 | 6.772 | 5.810 | .63 | .6009 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 10.644724 | 6.642 | 6.444 | .91 | .5856 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 9.651338 | 7.818 | 6.797 | 1.27 | .5744 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 10.861273 | 8.078 | 5.852 | .74 | .6549 | I |

| MON | DAY | WINDSPEED | WAVEHT | WAVEPD | HEAVE | PROPUL | DRIFT |
|---|---|---|---|---|---|---|---|
| 8 | 15 | 11.092415 | 5.331 | 6.050 | .57 | .5911 | I |

# APPENDIX C

# IMPORTANT ALGORITHMS

C1

```
   PROGRAM
SRTGEO(INPUT,OUTPUT,TTY,TAPE11=INPUT,TAPE12=OUTPUT)
   REAL SIGO,TEMP,HITE,HGTT,SWH,A,B,S,WG,WC,SIGMA,DAY,SEC
   REAL TIME,MSECSEC,LLONG,LLAT,LONG,LAT
C**********
C*** PROGRAM TO READ DATA FROM WINDOW OUTPUT FORM
C*** AND CONVERT TO BE READ AS INPUTS TO PROGRAM
C*** GEOAVHRR TO PLOT ALTIMETER TRACKS ON THERMAL IMAGE
C**********
C
C   MAIN LOOP FOR READING THE DATA FROM FILE
C
C*** LOOP INPUT DATA UNTIL END OF FILE
   DAY=1
   WHILE(DAY.NE.0) DO
C
C***    READ WINDOW DATA IN FREE FORMAT FORM
      READ(11,*)DAY,SEC,MSEC,LONG,LAT,HITE,SWH,SIGO
C
C***    TIME CALCULATION OF DAY SEC MSEC
       DAYSEC=0.
       IF(DAY.NE.0) THEN
           DAYSEC=(DAY-1)*86400.
       ENDIF
        MSECSEC=MSEC/1000000.
      TIME=DAYSEC+SEC+MSECSEC
C
C      LONG-LAT CALC.
        LLONG=LONG/1000000.
         LLAT=LAT/1000000.
C
C***    PROCESSING HGT TO METERS
        HGTT=HITE/100.
C
C***    WRITE IN FORMAT FOR GEOAVHRR INPUT FORM FOR TRACK
C***     MAPPING PURPOSES
       WRITE(12,30)TIME,LLONG,LLAT,HGTT
30        FORMAT(2X,F16.6,2X,F10.6,2X,F10.6,2X,F5.2)
   ENDWHILE
   STOP
   END
```

```
    PROGRAM
SRTAVG(INPUT,OUTPUT,TAPE11=INPUT,TAPE12=OUTPUT)
    REAL
SIGO,TEMP,HGT,HGTT,SWH,A,B,S,WG,WC,SIGMA,DAY,SEC,WW
    REAL
DAYSUM,SECSUM,HGTSUM,SWHSUM,SIGSUM,DAYAVG,SECAVG,HGTAV
G,SWHAVG
   REAL SIGAVG,N,DAYDEF,SECDEF
C**********
C***  PROGRAM TO READ INPUT FROM WINDOW PROGRAM OUTPUT
C***  THEN WILL DEFINE INDIVIDUAL TRACK ALONG DATA
C***  AVERAGE TRACK DATA FOR ANALYSIS
C**********
C
C    MAIN LOOP FOR READING THE DATA FROM FILE
C
C***  DEFINE INITIAL VALUES
      DAY=1.0
      DAYSUM=0.0
      SECSUM=0.0
      HGTSUM=0.0
      SWHSUM=0.0
      SIGSUM=0.0
      N=0.0
C
C***  READ INPUT UNTIL END OF FILE IS REACHED
    WHILE(DAY.NE.0) DO
C    NEW TRACK
C
C***      READ INPUT IN FREE FORMAT
100      READ(11,*)DAY,SEC,MSEC,LONG,LAT,HGT,SWH,SIGMA
C
C***      DEFINE DIFFERENCE VALUES FOR A NEW TRACK
        IF(N.EQ.0) THEN
           DAY2=DAY
           SEC2=SEC
        ENDIF
C
C***      CALCULATE DIFFERENCE
        DAYDEF=ABS(DAY-DAY2)
        SECDEF=ABS(SEC-SEC2)
C
C***      USE DIFFERENCE VALUES TO LOCATE NEW TRACK
```

```
      IF((DAYDEF.NE.0).OR.(SECDEF.GE.600)) GO TO 200
C
C***    INCREMENT N BY 1 (COUNTER)
      N=N+1
C
C***    ADD VALUES OF A GIVEN TRACK
      DAYSUM=DAYSUM+DAY
      SECSUM=SECSUM+SEC
      HGTSUM=HGTSUM+HGT
      SWHSUM=SWHSUM+SWH
      SIGSUM=SIGSUM+SIGMA
C
C***    RESET DIFFERENCE VALUES FOR ANOTHER TRACK
      DAY2=DAY
      SEC2=SEC
      GO TO 100
C
C***    NEW TRACK DISCOVERED, AVERAGE VALUES IN OLD TRACK
200     DAYAVG=DAYSUM/N
      SECAVG=SECSUM/N
      HGTAVG=HGTSUM/N
      SWHAVG=SWHSUM/N
      SIGAVG=SIGSUM/N
C
C***    SET NEW SUM VALUE
      DAYSUM=DAY
      SECSUM=SEC
      HGTSUM=HGT
      SWHSUM=SWH
      SIGSUM=SIGMA
C
C***    PROCESSING HGT FROM CM TO METERS
      HGTT=HGTAVG/100.
C
C***    PROCESSING SWH FROM 0.1 METERS TO METERS
      SWHM=SWHAVG/10.
C
C***    PROCESSING SIGMA TO WC IN METERS/SECOND
C
      SIGO=SIGAVG/10.
      S=10**(-((SIGO+2.1)/10))
      IF(SIGO.GT.(10.9)) THEN
          A=0.01595
```

C4

```
          B=0.017215
        ELSE IF(SIGO.LE.(10.12)) THEN
          A=0.080074
          B=-0.124651
        ELSE
          A=0.03983
          B=-0.031996
        ENDIF
C
        WG=EXP((S-B)/A)
        IF(WG.GT.16) THEN
          WC=WG
        ELSE
          WW=(2.087799*WG)-
(0.3649928*WG**2)+(0.04062421*WG**3)
            WC=WW-(0.001904952*WG**4)+(0.00003288189*WG**5)
        ENDIF
C
C***     WRITE TO OUTPUT IN NEW FORMAT
        WRITE(12,30)DAYAVG,SECAVG,HGTT,SWHM,WC
30         FORMAT(F4.0,2X,F6.0,2X,F7.3,2X,F4.1,2X,F10.2)
C
C***     RESET DIFFERENCE VALUES AND COUNTER
        DAY2=DAY
        SEC2=SEC
        N=1
    ENDWHILE
    STOP
    END
```

C5

```
    PROGRAM
DEFWIN(INPUT,OUTPUT,TAPE11=INPUT,TAPE12=OUTPUT)
    INTEGER DAY,SEC,MSEC,LONG,LAT,HGT,SWH,SIGMA
C**********
C*** PROGRAM TO READ IN RAW ALTIMETER DATA AND TO
C*** PROCESS TO BE READ IN A FREE FORMAT
C*** A WINDOW IS DEFINED TO REDUCE DATA
C**********
C
C   MAIN PROGRAM
C
C*** LOOP THRU DATA UNTIL END OF FILE IS REACHED
    DAY=1
    WHILE(DAY.NE.0) DO
C***     READ FROM INPUT FILE THE FORMAT PROVIDED
10       READ(11,100)DAY,SEC,MSEC,LONG,LAT,HGT,SWH,SIGMA
100         FORMAT  (I3,I5,I6,2I9,I4,I3,I3)
C
C***     DEFINE WINDOW IN THE GULF
          IF((LAT.LE.272000000).AND.(LAT.GT.269000000))  THEN
            IF((LONG.GE.23000000).AND.(LONG.LE.28500000))  THEN

WRITE(12,200)DAY,SEC,MSEC,LONG,LAT,HGT,SWH,SIGMA
200              FORMAT(I3,2X,I5,2X,I6,2X,I9,2X,I9,2X,I4,2X,I3,2X,I3)
            ENDIF
        ENDIF
          IF((LAT.LE.276000000).AND.(LAT.GT.272000000))  THEN
            IF((LONG.GE.23000000).AND.(LONG.LE.29250000))  THEN

WRITE(12,200)DAY,SEC,MSEC,LONG,LAT,HGT,SWH,SIGMA
            ENDIF
        ENDIF
    ENDWHILE
    STOP
    END
```

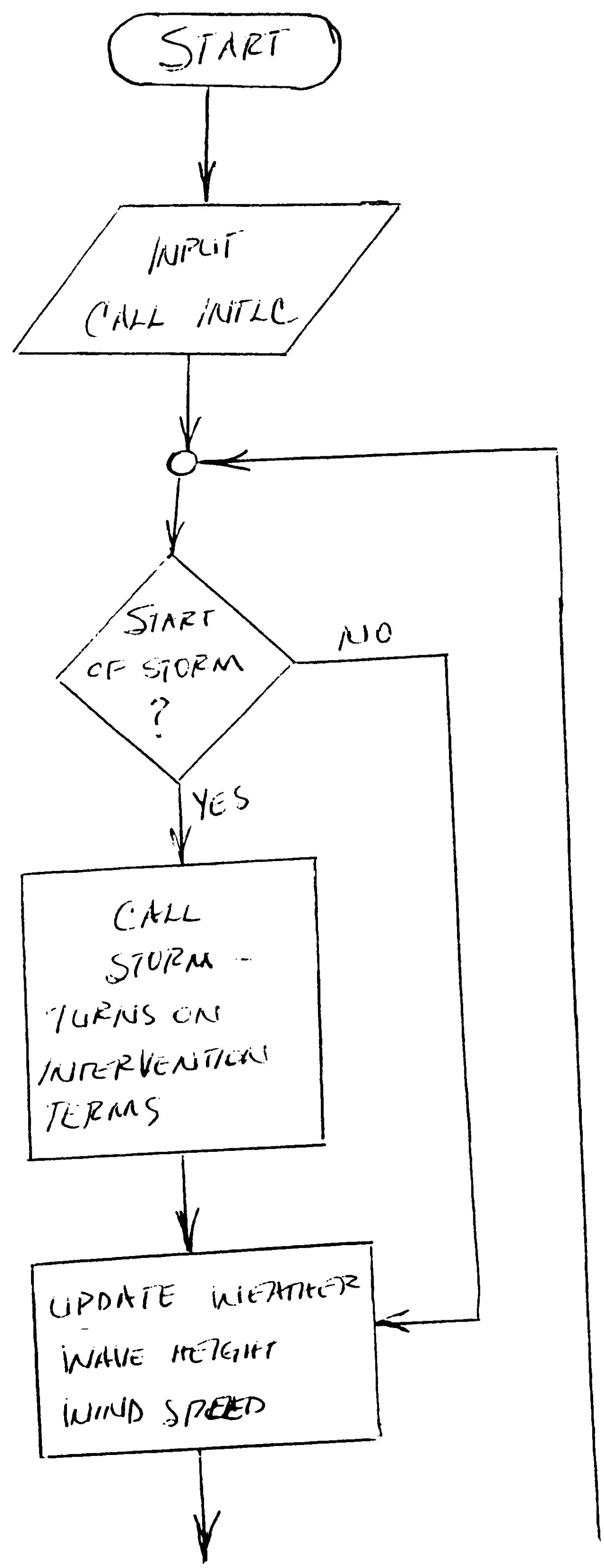
# APPENDIX D

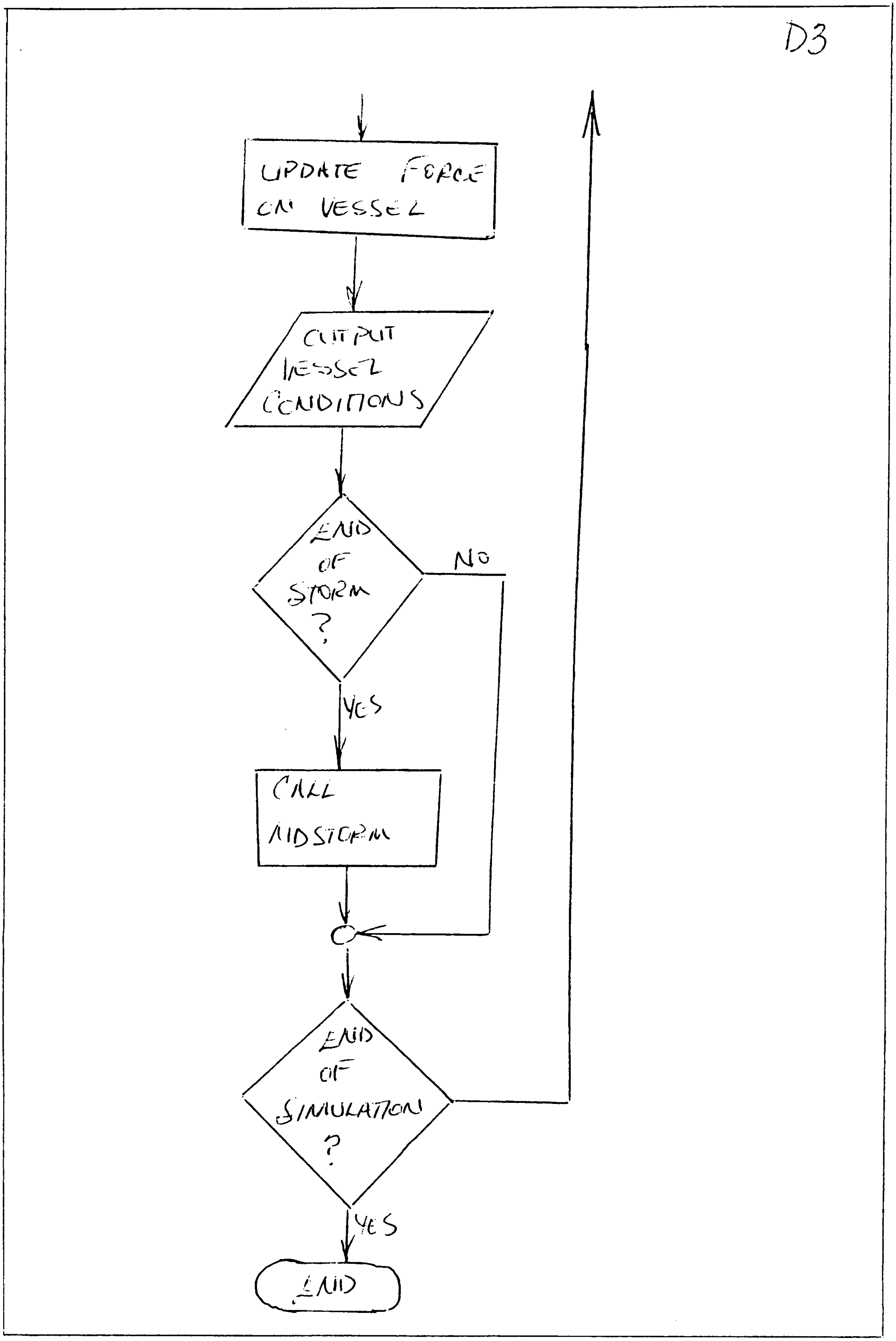
# FLOWCHART OF COMPLETE PROGRAM

THIS SECTION PROVIDES A FLOW CHART OF THE PROGRAM TO SHOW THE GENERAL OPERATION OF THE PROGRAM.

FOR A MORE DETAILED LOOK AT THE SPECIFIC WORKINGS OF THE PROGRAM, SEE APPENDIX E: LISTING OF COMPUTER PROGRAM.

D2


START
INPUT
CALL INTLC
START
OF STORM
?
NO
YES
CALL
STORM -
TURNS ON
INTERVENTION
TERMS
UPDATE WEATHER
WAVE HEIGHT
WIND SPEED




ORIGINAL PAGE IS
OF POOR QUALITY

# APPENDIX E

# LISTING OF COMPUTER PROGRAM

```
      PROGRAM
MAIN(INPUT,TAPE1,TAPE2,TAPE3,TAPE4,TTY,TAPE5=TTY,
     $TAPE7,TAPE8,TAPE9,TAPE10,OUTPUT,TAPE11=INPUT,
     $TAPE12=OUTPUT,TAPE13,TAPE14,TAPE15,TAPE16)
      COMMON/BRY/ROLD,RNEW,TEMP
      COMMON/SCOM1/ TNEXT,TNOW,TAB(23)
      COMMON/UCOM1/
HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD,IV(3)

1,WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(12,16)
      COMMON/UCOM3/ IPP,JQQ,KR,MS,NB
      COMMON/UCOM4/
MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
     1NSTORMS(12),MONTH,NHDAY,NDAY
      COMMON/TRFSERI/
KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(4),
     1SIG,RMU,TX(8),TY(4),TU(4),ADDON
      COMMON/TSERIES/DELTA(3),SIGMA(3),NSAMP(3),THETA(3,48)
     1,X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)
      COMMON/STORMTM/LSTART,LSTOP,LSTRM,STRM
      COMMON/FORS/FORWAV,FORWIN,FORCUR,FORNET,PROPUL,DRIFT
      REAL MEAN,VAR
      MEAN = 0.0
      VAR  = 1.0
      CALL INTLC
      WRITE(5,*) 'BEGIN SIMULATION'
      DO 10 I = 1,20
        IF (NHDAY.EQ.LSTART) THEN
            CALL STORM
            LSTRM = LSTRM + 1
        ENDIF
        CALL WEATHER
        CALL FORCE
        CALL REPORT
        IF (NHDAY.EQ.LSTOP) THEN
            CALL NDSTORM
        ENDIF
   10 CONTINUE
      WRITE(5,*) 'THERE WERE ',LSTRM,' STORMS'
   11 CONTINUE
      STOP
      END
```

```
      SUBROUTINE INTLC
C *** CALLED BY SLAM BEFORE EACH SIMULATION TO READ INPUT
DATA
C    SETS, SET INITIAL CONDITIONS, & SCHEDULE INITIAL EVENTS
       COMMON/TSERIES/DELTA(3),SIGMA(3),NSAMP(3),
       1THETA(3,48),X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)
      COMMON/BRY/ROLD,RNEW,TEMP
      COMMON/STORMTM/LSTART,LSTOP,LSTRM,STRM
       COMMON/TRFSERI/
KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(4),
     1SIG,RMU,TX(8),TY(4),TU(4),ADDON
      COMMON/SCOM1/TNEXT,TNOW,TAB(23)
       COMMON/UCOM1/HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
      COMMON/UCOM3/ IPP,JQQ,KR,MS,NB
      COMMON/UCOM4/
MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
     1NSTORMS(12),MONTH,NHDAY,NDAY
      DIMENSION FEE(2),THE(2),DEL(2),OMEGA(3)
      LSTRM = 0
      NHDAY = 0
      NNRUN = 1
      TNOW  = 0
C *** INITIALIZE ARRAYS FEE,THE,DEL,OMEGA TO 0
      DO 1 I = 1,2
         FEE(I) = 0.0
         THE(I) = 0.0
         DEL(I) = 0.0
    1 CONTINUE
      DO 2 I = 1,3
         OMEGA(I) = 0.0
    2 CONTINUE
C *** INITIALIZE RANDOM NUMBER GENERATOR
C    GENERATE FIRST 1000 FIBONACCI NUMBERS
      ROLD = 0.0
       RNEW = 0.00001
       DO 10 I=1,1000
         TEMP = RNEW
         RNEW = RNEW + ROLD
         ROLD = TEMP
```

```
        IF (RNEW.GE.1.0) THEN
            RNEW = RNEW - 1.0
        ENDIF
 10 CONTINUE
C *** READ RAO'S
      READ (9,170) (TAB(I),I=1,23)
 170 FORMAT (F4.2)
C *** READ INITIAL SUPPLIES ON DRILLING VESSEL
      READ(8,*) (RINIT(I),I=1,6)
C *** INITIALIZE INTERVENTION TERMS
      DO 20 I=1,3
  20 IV(I)=0.
     IF(NNRUN.NE.1) GO TO 610
C *** READ EMPIRICAL STORM DISTRIBUTIONS
      READ(8,*) (MSTORMS(I),I=1,12),(NSTORMS(I),I=1,12)
      READ(8,*) ((PR(I,J),J=1,25),I=1,12)
      READ(8,*) ((TTNS(I,J),J=1,25),I=1,12)
      READ(8,*) ((CLS(I,J),J=1,2),I=1,12)
      READ(8,*) ((RLENGTH(I,J),J=1,16),I=1,12)
      READ(8,*) ((STIME(I,J),J=1,16),I=1,12)
C *** READ DATE
    READ(8,*) MONTH,NDAY
C *** READ WAVE HEIGHT AND WAVE PERIOD INTERVENTION TERMS
      READ(10,*) (WH(I),I=1,3),(WP(J),J=1,3)
 195 FORMAT (F8.4)
     DO 60 KS=1,2
C *** READ EACH ARMA MODEL
    READ(10,*) IP(KS),JQ(KS),DELTA(KS),SIGMA(KS),NSAMP(KS)
 200 FORMAT (I1,1X,I1,1X,F6.3,F5.3,1X,I3)
 210 FORMAT (9(1X,F5.3))
C *** READ ARMA PARAMETERS
      IF (IP(KS).GT.0) READ (10,*) (PHI(KS,I),I=1,IP(KS))
    IF (JQ(KS).GT.0) READ (10,*) (THETA(KS,J),J=1,JQ(KS))
    YY=ARMA(0,KS)
    NN=.10*NSAMP(KS) + 2*(IP(KS)+JQ(KS))
C *** GENERATE FIRST NN VALUES.
    DO 30 K=1,NN
 30 YY=ARMA(1,KS)
 60 CONTINUE
C *** READ TRANSFER FUNCTION PARAMETERS
    READ(10,*) IPP,JQQ,KR,MS,NB,SIG,RMU
 220 FORMAT (I1,1X,I1,1X,I1,1X,I1,1X,I1,1X,F6.3,1X,F6.3)
     IF(IPP.GT.0) READ (10,*) (FEE(I),I=1,IPP)
```

```
      IF(JQQ.GT.0) READ (10,*) (THE(I),I=1,JQQ)
      IF(KR.GT.0) READ (10,*) (DEL(I),I=1,KR)
      READ(10,*) (OMEGA(I),I=1,MS+1)
  230 FORMAT (3(1X,F5.3))
C *** TWICE AS MANY HALF DAYS(NHDAY) AS DAYS
  610 NHDAY=2*NDAY
C *** A, B, C ARRAYS ARE PARAMETERS MULTIPLIED BY PAST
SERIES'
C    VALUES IN TRANSFER FUNCTION GENERATION & INTERVENTION
C    CALCULATIONS. THEY ARE DERIVED FROM TRANSFER FUNCTION
C    INPUT PARAMETERS
      DO 240 I=1,4
      A(I)=0.0
      B(I)=0.0
      C(I)=0.0
  240 CONTINUE
      B(5)=0.
      A(1)=-FEE(1)-DEL(1)
      A(2)=-DEL(2)+DEL(1)*FEE(1)-FEE(2)
      A(3)=DEL(2)*FEE(1)+DEL(1)*FEE(2)
      A(4)=DEL(2)*FEE(2)
      A(5)=(1-DEL(1)-DEL(2))*(1-FEE(1)-FEE(2))
      B(1)=OMEGA(1)
      B(2)=-OMEGA(1)*FEE(1)-OMEGA(2)
      B(3)=-OMEGA(1)*FEE(2)+OMEGA(2)*FEE(1)-OMEGA(3)
      B(4)=OMEGA(2)*FEE(2)+FEE(1)*OMEGA(3)
      B(5)=OMEGA(3)*FEE(2)
      C(1)=-DEL(1)-THE(1)
      C(2)=-DEL(2)+THE(1)*DEL(1)-THE(2)
      C(3)=THE(1)*DEL(2)+DEL(1)*THE(2)
      C(4)=DEL(2)*THE(2)
      SUM=1.0
      IF(IP(1).EQ.0) GO TO 248
C *** CALCULATE MEAN(XMU) OF WIND SERIES
      DO 245 I=1,IP(1)
  245 SUM=SUM-PHI(1,I)
  248 XMU=DELTA(1)/SUM
C *** DETERMINE WHAT MUST BE ADDED ON TO TRANSFER FUNCTION
TO
C    BRING TO MEAN
      ADDON=(1-FEE(1)-FEE(2))*((1-DEL(1)-DEL(2))*RMU-(OMEGA(1)-
OMEGA(2)
     X-OMEGA(3))*XMU)
```

```
      MAXA=MAXB=MAXC=0
C *** DETERMINE MAX A, B, C ELEMENTS > 0
  620 DO 500 I=1,4
      IF(A(I).NE.0.0) MAXA=I
      IF(B(I).NE.0.0) MAXB=I
  500 IF(C(I).NE.0.0) MAXC=I
      IF(B(5).NE.0.0) MAXB=5
C *** INITIALIZE TRANSFER FUNCTION--BRING TO STEADY STATE
  630 YY=TRANSFR(0)
      NN=30+2*(IPP+JQQ)
C *** GENERATE ENOUGH OF TRANSFER FUNCTION PROCESS TO BRING
TO
C     TO STEADY STATE
      DO 510 K=1,NN
      WIND=ARMA(1,1)
  510 YY=TRANSFR(1)
      IF(NNRUN.NE.1) GO TO 640
C *** WRITE ECHO REPORT FOR ALL INPUT VARIABLES
      WRITE(12,300)
      WRITE(12,641)
      DO 1020 I=1,2
      WRITE(12,650)
      WRITE(12,660) I,IP(I),JQ(I),DELTA(I),SIGMA(I),NSAMP(I)
      IF(IP(I).GT.0) WRITE(12,670) (PHI(I,J),J=1,IP(I))
 1020 IF(JQ(I).GT.0) WRITE(12,680) (THETA(I,J),J=1,JQ(I))
      WRITE(12,690)
      WRITE(12,700)
      DO 1030 I=1,3
 1030 WRITE(12,710) WH(I),WP(I)
      WRITE(12,720)
      WRITE(12,730)
      WRITE(12,740) IPP,JQQ,KR,MS,NB,SIG,RMU
      IF(IPP.GT.0) WRITE(12,670) (FEE(I),I=1,IPP)
      IF(JQQ.GT.0) WRITE(12,680) (THE(I),I=1,JQQ)
      IF(KR.GT.0) WRITE(12,750) (DEL(I),I=1,KR)
      WRITE(12,760) (OMEGA(I),I=1,MS+1)
      WRITE(12,770)
      DO 1040 I=1,12
 1040 WRITE(12,780) (MSTORMS(I),NSTORMS(I),I)
      DO 1050 I=1,12
      WRITE(12,790) I
      WRITE(12,800) (PR(I,J),J=1,10),(TTNS(I,J),J=1,10)
      WRITE(12,800) (PR(I,J),J=11,20),(TTNS(I,J),J=11,20)
```

```
1050 WRITE(12,810) (PR(I,J),J=21,25),(TTNS(I,J),J=21,25)
     DO 1055 I=1,12
     WRITE(12,820) I
1055 WRITE(12,830) (RLENGTH(I,J),J=1,16),(STIME(I,J),J=1,16)
     WRITE(12,840)
     WRITE(12,850)
     DO 1060 I=1,12
1060 WRITE(12,860) (I,CLS(I,1),CLS(I,2))
     WRITE(12,870) MONTH,NDAY
 300 FORMAT(28X,'INPUT ECHO REPORT')
 641 FORMAT(//8X'INPUT FOR ARMA MODELS')
 650 FORMAT(2X'MODEL',3X,'P',5X,'Q',3X,'DELTA',1X,'SIGMA',1X,
    X'NSAMP')
 660 FORMAT(4XI1,5XI1,5XI1,2X,F6.3,1X,F5.3,I6)
 670 FORMAT(' PHI S='(9F6.3))
 680 FORMAT(' THETA S='(9F6.3))
 690 FORMAT(//' INTERVENTION WEIGHTS')
 700 FORMAT(5X,'HEIGHT',4X,'PERIOD')
 710 FORMAT(1X,2(3XF7.4))
 720 FORMAT(//' TRANSFER FUNCTION MODEL INPUT')
 730 FORMAT(' NOISE P',2X,'NOISE Q',2X,'OUTPUT ORDER',2X,
    X'INPUT ORDER',2X,'INPUT BACKSHIFT',2X,'SIGMA',2X,'MEAN')
 740 FORMAT(4X,I1,8X,I1,12X,I1,13X,I1,13X,I1,8X,F5.3,1X,F5.3)
 750 FORMAT(' DELTA S='(2F6.3))
 760 FORMAT(' OMEGA S='(3F6.3))
 770 FORMAT(//' NO. OF TIME BETWEEN STORMS STORM LENGTHS
FOR MONTH')
 780 FORMAT(16XI2,16XI2,12X,I2)
 790 FORMAT(//' CDF CUTOFFS & TIME BETWEEN STORMS FOR
MONTH',I3)
 800 FORMAT(1X(10F5.2)/1X(10F5.0))
 810 FORMAT(1X(5F5.2)/1X(5F5.0))
 820 FORMAT(//' CDF CUTOFFS & STORM LENGTHS FOR MONTH',I3)
 830 FORMAT(1X(16F4.2)/1X(16F4.0))
 840 FORMAT(//' CDF CUTOFFS FOR STORM CLASS')
 850 FORMAT(/' MONTH',2X,'1',4X,'2')
 860 FORMAT(2XI2,2X,(2F5.2))
 870 FORMAT(//' STARTING MONTH IS',I3,1X,'AND DAY IS',I3)
C *** PRINT STATE VARIABLES
 640 CONTINUE
C *** CALL FIRST WEATHER EVENT
     CALL WEATHER
C *** TBT CHOOSES TIME UNTIL FIRST STORM
```

```
     STRM=TBT(N)
     SLENGTH = 2
     LSTART = STRM
      WRITE(5,*)'STORM ',STRM
      WRITE(5,*)'LSTART ',LSTART
     RETURN
     END


     SUBROUTINE WEATHER
C *** UPDATE WEATHER MODEL--WAVE PERIOD, WAVE HEIGHT, AND
SHIP
C    HEAVE.  ALSO UPDATE VARIOUS SUPPLY USAGE RATES.
C    ALSO UDATE DATE.
      COMMON/SCOM1/ TNEXT,TNOW,TAB(23)
      COMMON/UCOM1/ HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
     COMMON/UCOM3/ IPP,JQQ,KR,MS,NB
      COMMON/UCOM4/
MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
    1NSTORMS(12),MONTH,NHDAY,NDAY
      COMMON/TSERIES/DELTA(3),SIGMA(3),NSAMP(3),
       1THETA(3,48),X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)
      COMMON/TRFSERI/
KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(4),
    1SIG,RMU,TX(8),TY(4),TU(4),ADDON
     COMMON/STORMTM/LSTART,LSTOP,LSTRM,STRM
     DIMENSION IVECTOR(3,4)
C *** FIRST TIME THROUGH INITIALIZE VARIABLES:
C    IIOLD: POINTS TO 'OLDEST'(J) INTERVENTION TERMS(IV) IN
IVECTOR
C    IVECTOR(I,J):J TH INTERVENTION TERM FOR CLASS I STORM
     IF(TNOW.GT.0.) GO TO 1
      IIOLD=4
      DO 10 I=1,3
      DO 10 J=1,4
   10 IVECTOR(I,J)=0
C *** IF NEW MONTH, MUST UPDATE NHDAY(# OF 1/2 DAYS) &
MONTH
    1 IF(NHDAY.LT.61) GO TO 5
     NHDAY=1
```

```
      MONTH=MONTH+1
      LSTOP = SLENGTH + LSTART - 61
      STRM = TBT(N)
      LSTART = STRM
      IF(MONTH.EQ.13) MONTH=1
C *** UPDATE ARMA(P,Q) WIND SPEED MODEL.
    5 WIND=ARMA(1,1)
C *** REDRAW WIND IF < OR = 0.
      IF(WIND.LT.0.) GO TO 5
      REPEAT=0.
C *** EVALUATE WAVE PERIOD INTERVENTION MODEL
   35 PERIOD=ARMA(1,2)+WP(1)*IV(1)+WP(2)*IV(2)+WP(3)*IV(3)
C *** IF JUST REDRAWING PERIOD, DON'T UPDATE HEIGHT
      IF(REPEAT.EQ.1.0) GO TO 360
C *** EVALUATE WAVE HEIGHT TRANSFER FUNCTION MODEL
   38 HEIGHT=TRANSFR(1)+WH(1)*IV(1)+WH(2)*IV(2)+WH(3)*IV(3)
C *** FIND PAST INTERVENTION TERMS BY GOING THROUGH IVECTOR
ARRAY
C    START WITH MOST RECENT TO 'OLDEST'
C    EFFECTS OF PAST INTERVENTION TERMS MUST BE ACCOUNTED
FOR
C    WHEN UPDATING WEATHER PROCESSES WITH INTERVENTION
TERMS
  360 DO 100 II=1,4
      I=MOD(IIOLD+II,4)
      IF(I.EQ.0) I=4
C *** IF JUST REDRAWING PERIOD, DON'T UPDATE PERIOD
      IF(REPEAT.EQ.1.0) GO TO 370
C *** ADD TO HEIGHT IVECTOR*CORRESPONDING A ELEMENTS
      HEIGHT=HEIGHT+A(II)*(IVECTOR(1,I)*WH(1)+IVECTOR(2,I)*
     XWH(2)+IVECTOR(3,I)*WH(3))
C *** IF JUST REDRAWING HEIGHT, DON'T UPDATE PERIOD
      IF(REPEAT.EQ.2.0) GO TO 100
C *** DON'T INCLUDE VALUES PAST THOSE NECESSARY
  370 IF(II.GT.IP(1)) GO TO 100
C *** SUBTRACT FROM PERIOD IVECTOR*CORRESPONDING PHI
ELEMENTS
      PERIOD=PERIOD-PHI(2,II)*(IVECTOR(1,I)*WP(1)+IVECTOR(2,I)*
     XWP(2)+IVECTOR(3,I)*WP(3))
  100 CONTINUE
C *** IF PERIOD < OR = 0., REDRAW & INDICATE BY SETTING
C    REPEAT
      IF(PERIOD.GT.0.) GO TO 385
```

```
      REPEAT=1.
      GO TO 35
C *** IF HEIGHT < OR = 0., REDRAW HEIGHT & INDICATE BY
C    SETTING REPEAT VARIABLE
  385 IF(HEIGHT.GT.0.) GO TO 390
      REPEAT=2.
      GO TO 38
C *** FOR EACH CLASS STORM PUT CURRENT INTERVENTION TERM
C    VALUE WHERE "OLDEST" ELEMENT HAD BEEN IN IVECTOR FOR
C    THAT CLASS STORM
  390 DO 200 I=1,3
  200 IVECTOR(I,IIOLD)=IV(I)
C *** UPDATE IIOLD, WHERE IIOLD IS BETWEEN 1 & 4
       IIOLD=IIOLD-1
       IF(IIOLD.EQ.0) IIOLD=4
C *** DETERMINE DRILL SHIP'S HEAVE RESPONSE, THROUGH USE OF
C    BRETSCHNEIDER'S SPECTRUM.  NUMERICALLY INTEGRATE HEAVE
C    SPECTRAL DENSITY BY TRAPEZOIDAL RULE.  USE FUNCTION
C    ZETA TO EVALUATE HEAVE SPECTRAL DENSITY
(BRETSCHNEIDER'S
C    SPECTRUM*RAO**2) AT ALL POSSIBLE FREQUENCIES
C    HEAVE IS SQUARE ROOT OF INTEGRAL OF HEAVE SPECTRAL
DENSITY
      SUM=0.
       DO 40 I=1,22
        N=I
          FREQ=4.*3.14159/(I+13)
      ZET1=ZETA(FREQ,N)
       F1=FREQ
        N=I+1
          FREQ=4.*3.14159/(I+14)
   40 SUM=SUM+(ZET1+ZETA(FREQ,N))*(F1-FREQ)/2.
     HEAVE=SQRT(SUM)
C *** UPDATE NHDAY(# OF HALF DAYS)
     NHDAY=NHDAY+1
     RETURN
     END


     SUBROUTINE NDSTORM
C *** END STORM AND SCHEDULE NEXT STORM
      COMMON/SCOM1/ TNEXT,TNOW,TAB(23)
      COMMON/UCOM1/ HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD
```

```
1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
      COMMON/UCOM4/
MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
    1NSTORMS(12),MONTH,NHDAY,NDAY
       COMMON/TSERIES/ DELTA(3),SIGMA(3),NSAMP(3),
        1THETA(3,48),X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)

COMMON/TRFSERI/KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(
4),
     1SIG,RMU,TX(8),TY(4),TU(4),ADDON
     COMMON/UCOM3/IPP,JQQ,KR,MS,NB
     COMMON/STORMTM/LSTART,LSTOP,LSTRM,STRM
C *** TURN STORM OFF BY SETTING INTERVENTION TERMS,IV, TO 0.
      DO 10 I=1,3
    10 IV(I)=0.
C *** FUNCTION TBT DETERMINES TIME UNTIL NEXT STORM
      STRM=TBT(N)
     LSTART = NHDAY + STRM
     RETURN
     END


     SUBROUTINE STORM
C *** THIS SUBROUTINE STARTS STORMS, DETERMINES THEIR CLASS
&
C    LENGTH FROM MONTHLY CUMULATIVE PDF'S.  CALLS S.
NDSTORM TO END
C    STORM
      COMMON/SCOM1/ TNEXT,TNOW,TAB(23)
       COMMON/UCOM1/ HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
      COMMON/UCOM4/
MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
    1NSTORMS(12),MONTH,NHDAY,NDAY
       COMMON/TSERIES/DELTA(3),SIGMA(3),NSAMP(3),THETA(3,48),
       1X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)

COMMON/TRFSERI/KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(
4),
```

```
     1SIG,RMU,TX(8),TY(4),TU(4),ADDON
     COMMON/UCOM3/IPP,JQQ,KR,MS,NB
     COMMON/STORMTM/LSTART,LSTOP,LSTRM,STRM
C *** CHOOSE PROB, A RANDOM NUMBER TO DETERMINE STORM
CLASS FROM
C    CUMULATIVE PDF FOR MONTH
     PROB=UNIFORM(I)
C *** IF PROB IS <= CLS(MONTH,1), THEN STORM IS CLASS 1, IF NOT,
C    CHECK IF CLASS 2 OR CLASS 3 STORM
     IF(PROB.GT.CLS(MONTH,1)) GO TO 10
C *** SINCE STORM IS CLASS 1 SET CORRESPONDING INTERVENTION
C    TERM TO 1,IV(1),TO TURN STORM ON
      IV(1)=1
     GO TO 30
C *** SEE IF STORM IS CLASS 2
  10 IF(PROB.GT.CLS(MONTH,2)) GO TO 20
C *** TURN CLASS 2 STORM ON
      IV(2)=1
     GO TO 30
C *** IF STORM NOT CLASS 1 OR CLASS 2, MUST BE CLASS 3, SO TURN
C    CLASS 3 STORM ON
  20 IV(3)=1
C *** CHOOSE PROB, A RANDOM NUMBER.
  30 PROB=UNIFORM(I)
     DO 40 I=1,NSTORMS(MONTH)
C *** FIND WHERE PROB LANDS IN CUMULATIVE PDF FOR STORM
LENGTH(RLENGTH)
C    FOR MONTH
C *** STORM LENGTH(SLENGTH) IS CORRESPONDING ELEMENT IN
STIME ARRAY
     IF(PROB.GT.RLENGTH(MONTH,I)) GO TO 40
     SLENGTH=STIME(MONTH,I)
     LSTOP = NHDAY + SLENGTH
     RETURN
  40 CONTINUE
     RETURN
     END


     SUBROUTINE FORCE
C *** THIS ROUTINE CALCULATES THE NET ENVIRONMENTAL FORCE
C    ON THE PLATFORM AS THE SUM OF WAVE DRIFT FORCE, WIND
C    FORCE, AND CURRENT FORCE. ALL FORCES ARE ASSUMED TO
```

```
C    CONCURRENT AND ACTING IN THE SAME DIRECTION TO GIVE
C    A WORST CASE SCENARIO. THE NET PROPULSION REQUIRED
C    TO KEEP STATION IS THE NET FORCE VALUE IS EXPRESSED
C    AS A PERCENTAGE OF THE MAXIMUM AVAILABLE THRUST
(120000  LBS).
      COMMON/FORS/FORWAV,FORWIN,FORCUR,FORNET,PROPUL,DRIFT
        COMMON/UCOM1/HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
        VC  = 1.5
         MAX = 120000.0
       FORWAV = FDRIFT(HEIGHT)
      FORCUR = FCURNT(VC)
       FORWIN = WINDFC(WIND)
      FORNET = FORWAV + FORCUR + FORWIN
       PROPUL = FORNET/MAX
        IF (PROPUL.GT.1.0) DRIFT = 1.0
       RETURN
       END


      SUBROUTINE REPORT
C *** THIS ROUTINE PRINTS THE CONDITIONS ENCOUNTERED AT THE
PLATFORM
        COMMON/UCOM1/HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
         COMMON/UCOM4/MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2)
     1,NSTORMS(12),MONTH,NHDAY,NDAY
      COMMON/FORS/FORWAV,FORWIN,FORCUR,FORNET,PROPUL,DRIFT
       IF (NDAY.EQ.1) THEN
       WRITE(5,*)'MON DAY WINDSPEED WAVEHT WAVEPD HEAVE
PROPUL DRIFT'
       ENDIF

WRITE(12,10)MONTH,NDAY,WIND,HEIGHT,PERIOD,HEAVE,PROPUL,DRI
FT
1 0
FORMAT(1X,I2,2X,I2,1X,F9.6,1X,F6.31X,F6.3,1X,F5.2,1X,F6.4,1X,F3.1)
      RETURN
      END
```

```
   FUNCTION ZETA(FREQ,N)
C *** DETERMINE VALUE OF INTEGRAL AT DIFFERENT FREQUENCIES,
C   WHERE INTEGRAL IS BRETSCHNEIDER'S FUNCTION*RAO**2
C   OR WAVE SPECTRAL DENSITY.
    COMMON/SCOM1/ TNEXT,TNOW,TAB(23)
    COMMON/UCOM1/ HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
    COMMON/UCOM4/
MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
   1NSTORMS(12),MONTH,NHDAY
     COMMON/TSERIES/DELTA(3),SIGMA(3),NSAMP(3),THETA(3,48),
     1X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)

COMMON/TRFSERI/KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(
4),
   1SIG,RMU,TX(8),TY(4),TU(4),ADDON
   COMMON/UCOM3/IPP,JQQ,KR,MS,NB
    POWER=-1050./((PERIOD**4)*(FREQ**4))
C *** CALCULATE WAVE SPECTRAL DENSITY, SPECDF

SPECDF=4200.*(HEIGHT**2)*EXP(POWER)/((PERIOD**4)*(FREQ**5))
C *** TAB(N) IS RAO FOR GIVEN FREQUENCY, FREQ
   ZETA=SPECDF*(TAB(N)**2)
   RETURN
   END


   FUNCTION TBT(N)
C *** CALCULATE TIME UNTIL NEXT STORM, GIVEN MONTH
    COMMON/SCOM1/ TNEXT,TNOW,TAB(23)
    COMMON/UCOM1/ HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
    COMMON/UCOM4/
MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
   1NSTORMS(12),MONTH,NHDAY
     COMMON/TSERIES/DELTA(3),SIGMA(3),NSAMP(3),THETA(3,48),
```

```
      1X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)

COMMON/TRFSERI/KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(
4),
     1SIG,RMU,TX(8),TY(4),TU(4),ADDON
     COMMON/UCOM3/IPP,JQQ,KR,MS,NB
C *** CHOOSE PROB, A RANDOM NUMBER.
      PROB=UNIFORM(I)
      DO 10 I=1,MSTORMS(MONTH)
C *** FIND WHERE PROB LANDS IN CUMULATIVE PDF(PR) FOR MONTH
C *** TBT IS FOUND BY CORRESPONDING ELEMENT IN TTNS ARRAY
      IF(PROB.GT.PR(MONTH,I)) GO TO 10
      TBT=TTNS(MONTH,I)
      WRITE(5,*) 'PROB ',PROB
       WRITE(5,*)'PR(MONTH,I) ',PR(MONTH,I)
       WRITE(5,*)'TBT ',TBT
      RETURN
   10 CONTINUE
      RETURN
      END


      FUNCTION ARMA(IND,KS)
C *** GENERATE ARMA (P,Q) MODELS
C *** GENERATOR USES ARRAYS,X(SERIES) & U(WHITE NOISE SERIES),
C     TO ACCOUNT FOR DEPENDENT PAST VALUES. IOLD AND JOLD
POINT TO
C     THE OLDEST ELEMENT IN EACH ARRAY.  NEWEST ELEMENT IS
ONE ELE-
C     MENT OVER.
       COMMON/TSERIES/DELTA(3),SIGMA(3),NSAMP(3),
        1THETA(3,48),X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)
      COMMON/UCOM1/ HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
      COMMON/SCOM1/TNEXT,TNOW,TAB(23)

COMMON/TRFSERI/KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(
4),
     1SIG,RMU,TX(8),TY(4),TU(4),ADDON

COMMON/UCOM4/MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
```

```
    1NSTORMS(12),MONTH,NHDAY
     COMMON/UCOM3/IPP,JQQ,KR,MS,NB
C *** FIRST TIME THROUGH (IND=0), INITILIZE VARIABLES.
OTHERWISE,
C    GO TO 100 AND GENERATE SERIES.
      IF(IND.EQ.1) GO TO 100
       NIP=IP(KS)
    NJQ=JQ(KS)
     XMU = DELTA(KS)
      SUM = 1.0
C *** CALCULATE MAXIMUM LAG, LMAX
      LMAX = MAX0(NIP,NJQ)
C *** CALCULATE MEAN (XMU) OF SERIES
      IF (NIP .EQ. 0) GO TO 20
       DO 10 I=1,NIP
   10 SUM = SUM - PHI(KS,I)
   20 XMU = DELTA(KS)/SUM
C *** INITIALIZE OLDEST ELEMENT POINTERS, IOLD & JOLD, FOR
SERIES (X)
C    & WHITE NOISE SERIES (U) TO LAST ELEMENT IN EACH ARRAY
      IOLD(KS) = NIP
    JOLD(KS) = NJQ
      DO 30 LAG=1,LMAX
C *** INITIALIZE WHITE NOISE SERIES TO MEAN (0.)
      U(KS,LAG) = 0.0
C *** INITIALIZE SERIES (X,ARMA) TO MEAN (XMU)
   30 X(KS,LAG) = XMU
   35 ARMA = XMU
     RETURN
C *** WHITE NOISE (UO) IS NORMAL(0.,SIGMA)
  100 UO = ORMAL(0.0,SIGMA)
     ARMA = DELTA(KS) + UO
C *** IF ARMA NOT DEPENDENT ON PAST SERIES VALUES (X), DON'T
C    ADD THEM ON
C *** ARMA DEPENDS ON WHITE NOISE PLUS DELTA TO BRING SERIES
C    UP TO MEAN
      IF (IP(KS) .EQ. 0) GO TO 150
C *** GET PAST SERIES ELEMENTS (X) IN ORDER, FROM LAST TO
C    OLDEST
       DO 120 II=1,IP(KS)
      I = MOD(IOLD(KS)+II,IP(KS))
       IF (I.EQ.0) I=IP(KS)
C *** ADD TO ARMA PAST SERIES VALUES(X) TIMES PHI ARRAY
```

```
 120 ARMA = ARMA + PHI(KS,II)*X(KS,I)
C *** IF ARMA NOT DEPENDENT ON PAST WHITE NOISE VALUES(U),
C    DON'T ADD THEM ON
 150 IF (JQ(KS) .EQ. 0) GO TO 500
C *** GET PAST WHITE NOISE VARIABLES (U) FROM LAST PERIOD
C    TO OLDEST
     DO 170 JJ=1,JQ(KS)
    J = MOD(JOLD(KS)+JJ,JQ(KS))
     IF (J.EQ.0) J=JQ(KS)
C *** SUBTRACT PAST WHITE NOISE VARIABLES(U) TIMES THETA
ARRAY
 170 ARMA = ARMA - THETA(KS,JJ)*U(KS,J)
C *** IF ARMA IS DEPENDENT ON PAST SERIES VALUES (X), SAVE
C    ARMA WHERE OLDEST X ELEMENT IS.
 500 IF (IP(KS) .EQ. 0) GO TO 550
     X(KS,IOLD(KS)) = ARMA
C *** UPDATE IOLD WHERE IOLD IS BETWEEN 1 AND P
     IOLD(KS) = IOLD(KS) - 1
     IF (IOLD(KS) .EQ. 0) IOLD(KS) = IP(KS)
C *** IF ARMA NOT DEPENDENT ON PAST WHITE NOISE, DON'T
UPDATE
C    U ARRAY
 550 IF (JQ(KS) .EQ. 0) RETURN
C *** SAVE CURRENT WHITE NOISE (UO) WHERE OLDEST WHITE NOISE
C    HAD BEEN
    U(KS,JOLD(KS)) = UO
C *** UPDATE JOLD
     JOLD(KS) = JOLD(KS) - 1
     IF (JOLD(KS) .EQ. 0) JOLD(KS) = JQ(KS)
     RETURN
     END


     FUNCTION TRANSFR(IND)
C *** TRANSFR IS GENERATED TRANSFER FUNCTION VARIABLE.
C    TERM DEFINITIONS:
C    TY:PAST OUTPUT SERIES VALUES THAT CURRENT OUTPUT
VALUE DEPENDS ON
C    TX:INPUT SERIES VALUS THAT CURRENT OUTPUT VALUE
DEPENDS ON
C    TU:WHITE NOISE SERIES VALUES THAT CURRENT OUPUT VALUE
DEPENDS ON
```

```
C    A:CALCULATED IN S. INTLC FROM TRANSFER FUNCTION
PARAMETERS.
C      PARAMETERS TY SERIES IS MULTIPLIED BY TO GENERATE
TRANSFR.
C       LAST ELEMENT > 0. IS MAXA.
C    B:CALCULATED IN S. INTLC FROM TRANSFER FUNCTION
PARAMETERS.
C      PARAMETERS TX SERIES IS MULTIPLIED BY TO GENERATE
TRANSFR.
C       MAXB IS LAST ELEMENT > 0.
C    C:CALCULATED IN S. INTLC FROM TRANSFER FUNCTION
PARAMETERS.
C      PARAMETERS TU SERIES IS MULTIPLIED BY TO GENERATE
TRANSFR.
C       MAXC IS LAST ELEMENT > 0.
C *** THIS FUNCTION SAVES DEPENDENT VALUES IN TY, TX, & TU
ARRAYS
C    AND POINTS TO OLDEST ELEMENT WITH POINTERS.  NEWEST
C    ELEMENT IS TO RIGHT OF OLDEST.
      COMMON/SCOM1/ TNEXT,TNOW,TAB(23)
      COMMON/UCOM1/ HEAVE,HEIGHT,RINIT(6),IP(3),JQ(3),PERIOD

1,IV(3),WP(3),WH(3),WIND,COUNT,RLENGTH(12,16),SLENGTH,STIME(
12,16)
     COMMON/UCOM3/ IPP,JQQ,KR,MS,NB
      COMMON/TSERIES/DELTA(3),SIGMA(3),NSAMP(3),
      1THETA(3,48),X(3,60),U(3,60),IOLD(3),JOLD(3),PHI(3,48)
     COMMON/TRFSERI/
KOLD,LOLD,MOLD,MAXA,MAXB,MAXC,A(5),B(5),C(4),
    1SIG,RMU,TX(8),TY(4),TU(4),ADDON

COMMON/UCOM4/MSTORMS(12),PR(12,25),TTNS(12,25),CLS(12,2),
   1NSTORMS(12),MONTH,NHDAY
C *** IF FIRST TIME THROUGH, INITIALIZE VARIABLES AND RETURN.
C    OTHERWISE, GO TO LINE 100 AND GENERATE TRANSFR.
     IF(IND.EQ.1) GO TO 100
C *** INITIALIZE OLDEST ELEMENT POINTERS, KOLD, LOLD, & MOLD,
C    FOR OUTPUT ARRAY(TY), INPUT ARRAY (TX), & WHITE NOISE
ARRAY (TU),
C    AT LAST ELEMENT OF EACH.
    KOLD=MAXA
    LOLD=NB+MAXB
    MOLD=MAXC
```

```
C *** IF TRANSFR NOT DEPENDENT ON PAST OUTPUT SERIES (TY),
C    DON'T SAVE PAST OUTPUT VALUES
     IF(MAXA.EQ.0) GO TO 15
C *** INITIALIZE OUTPUT SERIES (TY) AT MEAN (RMU)
      DO 10 II=1,MAXA
   10 TY(II)=RMU
C *** CALCULATE MEAN (XMU) OF INPUT SERIES
   15 SUM=1.0
      IF(IP(1).EQ.0) GO TO 25
      DO 20 I=1,IP(1)
   20 SUM=SUM-PHI(1,I)
   25 XMU=DELTA(1)/SUM
C *** INITIALIZE INPUT ARRAY (TX) WITH MEAN (XMU)
      DO 30 II=1,MAXB
   30 TX(II+NB)=XMU
C *** IF TRANSFR NOT DEPENDENT ON PAST WHITE NOISE
VALUES(TU),
C    DON'T NEED TO SAVE PAST VALUES
     IF(MAXC.EQ.0) GO TO 50
C *** INITIALIZE WHITE NOISE ARRAY(TU) TO MEAN (0.)
      DO 40 II=1,MAXC
   40 TU(II)=0.
C *** INITIALIZE OUTPUT SERIES(TRANSFR AND HEIGHT) TO ITS
MEAN(RMU)
   50 TRANSFR=RMU
      HEIGHT=RMU
      RETURN
C *** IF TRANSFR NOT DEPENDENT ON PAST OUTPUT SERIES VALUES,
C    DON'T NEED TO UPDATA TY ARRAY
  100 IF(KR.EQ.0.AND.IPP.EQ.0) GO TO 110
C *** REPLACE OLDEST TY VARIABLE WITH LAST OUTPUT SERIES
VARIABLE
      TY(KOLD)=HEIGHT
C *** UPDATE KOLD, WHERE KOLD IS BETWEEN 1 AND MAXA
      KOLD=KOLD-1
      IF(KOLD.EQ.0) KOLD=MAXA
C *** REPLACE OLDEST TX VARIABLE WITH CURRENT INPUT SERIES
VARIABLE
  110 TX(LOLD)=WIND
C *** UPDATE LOLD, WHERE LOLD IS BETWEEN 1 AND NB+MAXB
      LOLD=LOLD-1
      IF(LOLD.EQ.0) LOLD=NB+MAXB
C *** CURRENT WHITE NOISE IS NORMAL(0.,SIGMA)
```

```
      UO=ORMAL(0.0,SIGMA)
C *** TRANSFR IS WHITE NOISE(UO) PLUS ADDON(BRINGS TRANSFR
C     UP TO MEAN)
      TRANSFR=UO+ADDON
C *** IF TRANSFER FUNCTION NOT DEPENDENT ON PAST OUTPUT
SERIES
C     VARIABLES, TY, DON'T NEED PAST VALUES
      IF(MAXA.EQ.0) GO TO 200
C *** FIND PAST OUTPUT SERIES VARIABLES, TY, STARTING FROM
C     LAST PERIOD TO OLDEST
      DO 120 II=1,MAXA
      I=MOD(KOLD+II,MAXA)
      IF(I.EQ.0) I=MAXA
C *** SUBTRACT FROM TRANSFR PAST OUTPUT SERIES VARIABLES,
TY,
C     TIMES CORRECT A TERMS
      TRANSFR=TRANSFR-A(II)*TY(I)
  120 CONTINUE
C *** FIND INPUT SERIES TERMS, TX, STARTING WITH CURRENT TERM
C     AND GOING BACK TO OLDEST
  200 DO 220 JJ=1,MAXB
      J=MOD(LOLD+JJ+NB,MAXB+NB)
      IF(J.EQ.0) J=MAXB+NB
C *** ADD TO TRANSFR B TERMS TIMES CORRECT TX TERMS
  220 TRANSFR=TRANSFR+B(JJ)*TX(J)
C *** IF OUTPUT SERIES NOT DEPENDENT ON PAST WHITE NOISE
VARI-
C     ABLES, DON'T ADD ON PAST TU TERMS
      IF(MAXC.EQ.0) GO TO 550
C *** FIND NEEDED PAST TU VARIABLES, STARTING WITH LAST TU,
C     THEN ONE BEFORE LAST, UNTIL REACH OLDEST
      DO 320 KK=1,MAXC
      K=MOD(MOLD+KK,MAXC)
      IF(K.EQ.0) K=MAXC
C *** ADD TU TERM TIMES CORRESONDING C ELEMENT TO TRANSFR
  320 TRANSFR=TRANSFR+C(KK)*TU(K)
C *** IF TRANSFER FUNCTION NOT DEPENDENT ON PAST WHITE
NOISE, DON'T
C     UPDATE TU ARRAY
  550 IF(MAXC.EQ.0) RETURN
C *** PUT UO(WHITE NOISE) WHERE OLDEST TU(WHITE NOISE) SERIES
HAD BEEN
      TU(MOLD)=UO
```

```
C *** UPDATE MOLD, WHERE MOLD MUST BE BETWEEN 1 AND MAXC
      MOLD=MOLD-1
      IF(MOLD.EQ.0) MOLD=MAXC
      RETURN
      END


      FUNCTION UNIFORM(N)
C *** THIS FUNCTION GENERATES A UNIFORMLY DISTRIBUTED
C     PSEUDORANDOM NUMBER BETWEEN 0 AND 1 FROM THE
C     FIBONACCI SEQUENCE. EVERY SECOND TERM IS USED TO
C     MAKE THE NUMBERS APPEAR MORE RANDOM.
      COMMON/BRY/ ROLD,RNEW,TEMP
       DO 10 I = 1,2
          TEMP = RNEW
          RNEW = RNEW + ROLD
          ROLD = TEMP
          IF (RNEW.GE.1.0) THEN
              RNEW = RNEW - 1.0
          ENDIF
   10 CONTINUE
      UNIFORM = RNEW
      RETURN
      END


      FUNCTION ORMAL(MEAN,VAR)
C *** THIS FUNCTION GENERATES A NORMALLY DISTRIBUTED
C     PSEUDORANDOM NUMBER FROM A NORMAL DISTRIBUTION
C     WITH MEAN "MEAN" AND VARIANCE "VAR" BY ADDING
C     12 UNIFORMLY DISTRIBUTED RANDOM NUMBERS BETWEEN
C     0 AND 1 AND SUBTRACTING 6.
      REAL MEAN,VAR,DEV,TEMP
       TEMP = 0.0
      DEV = SQRT(VAR)
       DO 10 I = 1,12
          TEMP = TEMP + UNIFORM(N)
   10 CONTINUE
       TEMP = TEMP - 6.0
      ORMAL = MEAN + (TEMP*DEV)
      RETURN
      END
```

```
      FUNCTION WINDFC(VW)
C *** THIS FUNCTION CALCULATES THE WIND FORCE ON THE
PLATFORM
C     ASSUMING A 60 FOOT DRAFT AND A 0 DEGREE HEADING.
      REAL VW,DW,CSCHA,CW
C *** VW IS THE VELOCITY OF THE WIND IN KNOTS
C     DW IS THE DIRECTION OF THE WIND IN RADIANS(0=NORTH)
C     CSCHA IS THE PRODUCT OF THE SHAPE COEFFICIENT, THE
C     HEIGHT COEFFICIENT, AND THE PROJECTED AREA IN SQUARE
C     FEET.
C       CW IS 0.0034 LB(FT**2)(KT**2)
       CW = 0.0034
       CSCHA = 19738.0
      FORCE = CW * CSCHA * (VW**2)
      WINDFC = FORCE
      RETURN
      END


      FUNCTION FDRIFT(WAVEHT)
C *** THIS FUNCTION READS WAVE DRIFT FORCE FROM AN ARRAY
C     BASED ON SEA STATE (SIGNIFICANT WAVE HEIGHT).
      REAL WAVEHT
      IF (WAVEHT.LE.2.9) THEN
          FDRIFT = 0.0
      ELSE IF (WAVEHT.LE.4.6) THEN
          FDRIFT = 5000.0
      ELSE IF (WAVEHT.LE.8.0) THEN
          FDRIFT = 17500.0
      ELSE IF (WAVEHT.LE.12.0) THEN
          FDRIFT = 25500.0
      ELSE IF (WAVEHT.LE.18.0) THEN
          FDRIFT = 41500.0
      ELSE IF (WAVEHT.LE.28.0) THEN
          FDRIFT = 58000.0
      ELSE
          FDRIFT = 73000.0
      ENDIF
      RETURN
      END


      FUNCTION FCURNT(VC)
```

```
C *** THIS FUNCTION CALCULATES THE FORCE ON THE PLATFORM
C    DUE TO OCEAN CURRENTSFROM THE FORMULA
C      CSS(CD*AC + CD*AP)VC^2
C    USING API RECOMMENDED DRAG COEFFICIENTS, THE CURRENT
C    FORCE IN LBS WAS DETERMINED BY BROWN & ROOT U.S.A.,INC.
C      TO BE 20077*VC^2
      REAL VC
      FCURNT = 20077.0 * VC**2
      RETURN
      END
```

# APPENDIX F

# USER'S GUIDE

F1

# USER'S GUIDE

The purpose of this guide is to help the user to operate the program. Access to the program, execution of the program, input, and output will be explained.

The program is located on the UT/Austin CDC Cyber system in account MEDC532. A backup is located in AGL account MEIE003. The program is saved under the name STATION.

To run the program on the Cyber, it must be read into a local file while under the TAURUS operating system. From the "period" prompt, type

**. READ STATION <cr>**

Next, the input files must be copied. There are three input files: ARINPUT, TABLE, and INITIAL. ARINPUT contains ARMA model parameters, TABLE contains platform RAO's, and INITIAL contains initial condition values. These files must be copied to TAPE10, TAPE9, and TAPE8. From the "period" prompt, type

**. READ ARINPUT = TAPE10 <cr>**
**. READ TABLE = TAPE9 <cr>**
**. READ INITIAL = TAPE8 <cr>**

Now, the program must be compiled using the Minnesota FORTRAN compiler. To compile the program, the edit buffer must be expanded using the RFL command, and all files must be rewound. A

I.
FZ.

slash must be typed at the end of the RFL command to delay the execution of this command until the next executable command is entered. From the "period" prompt, type

**. REWALLX**

**. RFL, 100000/ <cr>**

**. MNF, I=STATION <cr>**

All that remains is to run the program. From the "period" prompt, type

**. LGO <cr>**

The simulation results will be in file OUTPUT, along with a listing of the compiled program. To view the results, from the "period" prompt, type

**. SHOW OUTPUT**

A sample output is contained in Appendix B.

# APPENDIX G

# BIBLIOGRAPHICAL LISTING

# BIBLIOGRAPHY


1. Bedworth, David D. and James E. Bailey. Integrated Production Control Systems. John Wiley & Sons, New York, 1987.

2. Bowerman, Bruce L. and Richard T. O'Connell. Forecasting and Time Series. Duxbury Press, North Scituate, Massachusetts, 1979.

3. "Characterization of ARCO Sites 709 and 710 from Historical Data." Volume I - Currents and Hydrography, Science Applications, Inc, Raleigh, North Carolina, October 15, 1982.

4. "Collinear-Track Altimetry on the Gulf of Mexico from SEASAT: Measurements, Models, and Surface Truths." Journal of Geophysical Research, Vol 88, NO. C3. February, 28, 1983, pp.1625-36.

5. Crawford, Dr. Melba M. Time Series Models of Sea State Variables. Department of Mechanical Engineering, University of Texas at Austin, March, 1985.

6. Crawford, Dr. Melba M. Conversations throughout the semester. Department of Mechanical Engineering, University of Texas at Austin, Spring 1990.

7. Daily Weather Maps - February 27 through June 4, 1989. Climate Analysis Center, Washington, D.C. Obtained through Sonia Gallegos, Ph.D.

8. Ebbesmeyer, Curtis, editor. Frontal Eddy Dynamics (FRED) Experiment Off North Carolina. Volume II Technical Report. Evans-Hamilton, Seattle, 1989.

9. Elachi, Charles. Introduction to the Physics and Techniques of Remote Sensing. John Wiley & Sons, New York, 1987.

10. Evans, W.M., G.A. Futoma, and C.A. Lombana. "Performance Prediction of Floating Drilling Vessels for Various Operating Areas." Atlantic Richfield Company Journal of Petroleum Technology, 1978, pg 1688.

11. Greiner, W. and D. Tuturea. Preliminary Operability Analysis for Western Pacesetter III Conversion. Brown & Root U.S.A., Houston, October 25, 1989.

12. Glenn, Scott and George Forrestall. Observations of Gulf Stream Ring 83-E and Their Interpretation Using Feature Models. Shell Development Company, Houston, February 1, 1990.

13. Hoff, John C., PhD. A Practical Guide to Box-Jenkins Forecasting. Lifetime Learning Publications, Belmont, California, 1983.

14. Hoffman, Susan Elizabeth. An Integrated Model of Drilling Vessel Operations. Master's Thesis, University of Texas at Austin, December, 1982.

15. Lewis, Edward V. "The Motion of Ships in Waves." Principles of Naval Architecture, Chapter IX.

16. Lunderg, Dr. John. Conversations throughout the semester. Department of Aerospace Engineering, University of Texas at Austin, Spring 1990.

17. Moran, David. Conversation on March 22, 1990. David Taylor Labs, Bethesda, Maryland.

18. NASA/USRA University Advanced Design Program. Handbook for Students, 1988-89.

19. "Oceanographic Field Data Over the Russel Tracts 709 and 710 Volume I -Summer 1982 Over-the-Side Measurements." Science Applications, Inc, Raleigh, North Carolina, June 17, 1983.

20. Oceanography from Space. Joint Oceanographic Institutions Incorporated, Washington, D.C., July, 1984.

21. O'Keefe, Shawn. An Automated Method for Estimating Oceanic Flow Fields from Satellite Imagery. Doctoral Dissertation, University of Texas at Austin, December, 1989.

22. Richards, John A. Remote Sensing Digital Analysis. Springer-Verlag, Berlin Heidelburg, Germany, 1986.

23. Schaudt, Kenneth J. Meeting on March 14, 1990. Marathon Oil Company, Houston, Texas.

24. Torfason, Louis. Meeting on March 28, 1990. University of Texas at Austin, Department of Mechanical Engineering.

25. Tracks. Program available at University of Texas Center for Space Research Imaging Lab. Department of Mechanical Engineering, University of Texas at Austin.

26. Webb, David L. (Donhaiser Marine Inc) and William P. Schneider (University of Houston). "Motion Response and Station Keeping Theoretical/Test Correlation of Glomar Explorer." Offshore Technology Conference, 1980.

# APPENDIX H

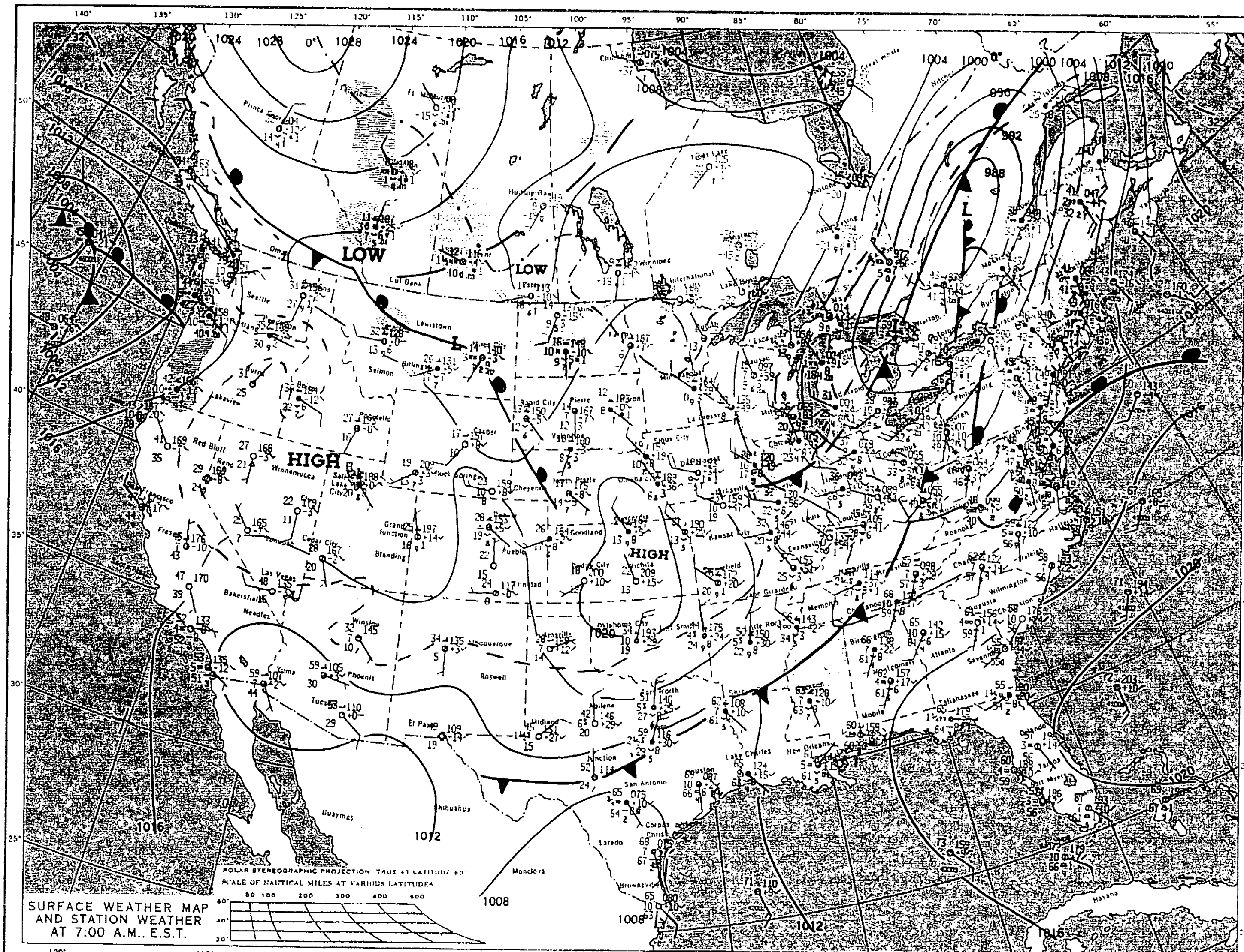
# WEATHER MAP

ORIGINAL PAGE IS
OF POOR QUALITY

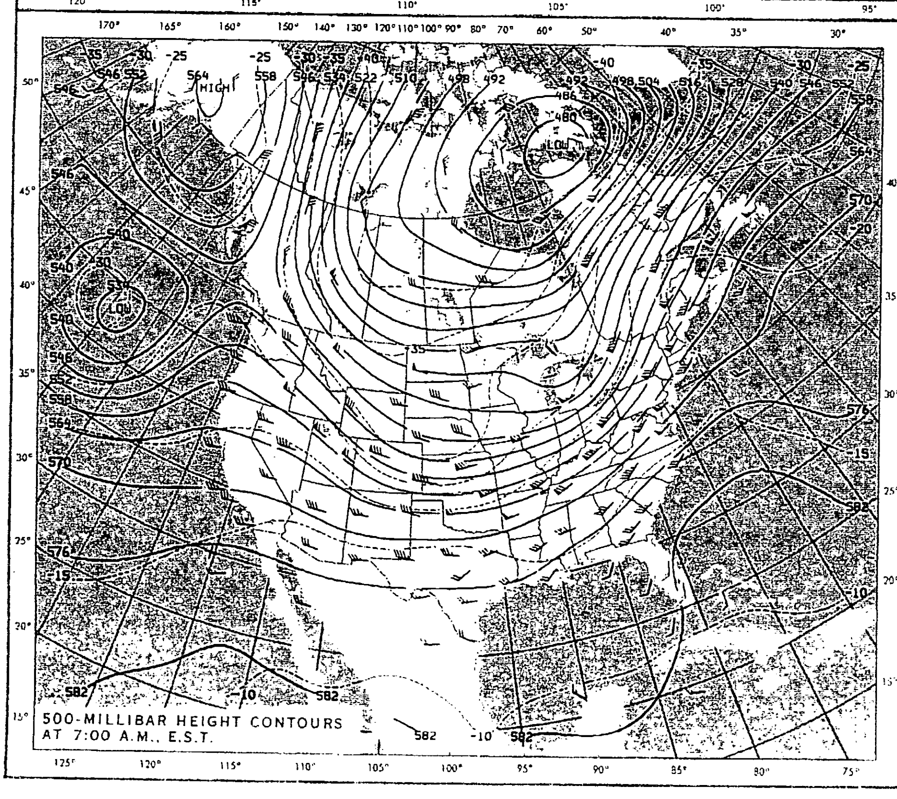
ORIGINAL PAGE
COLOR PHOTOGRAPH

H1

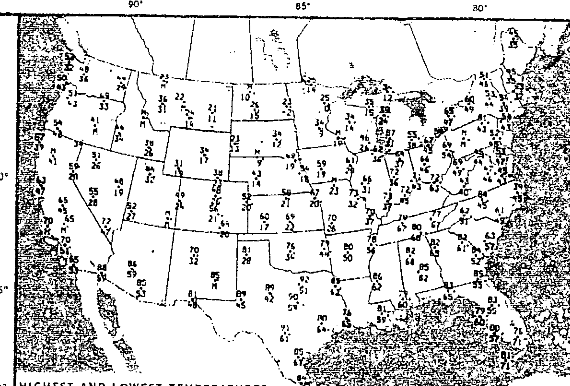
WEDNESDAY, MARCH 15, 1989

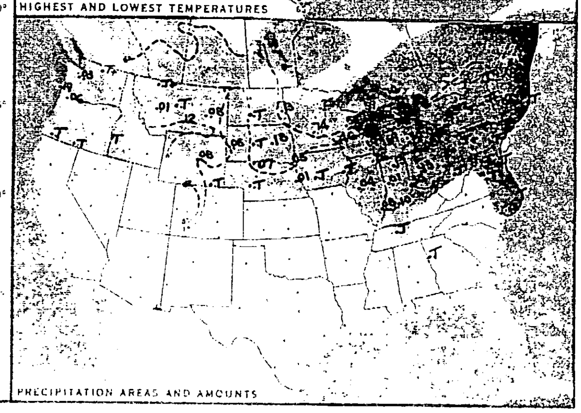


SURFACE WEATHER MAP AND STATION WEATHER AT 7:00 A.M., E.S.T.



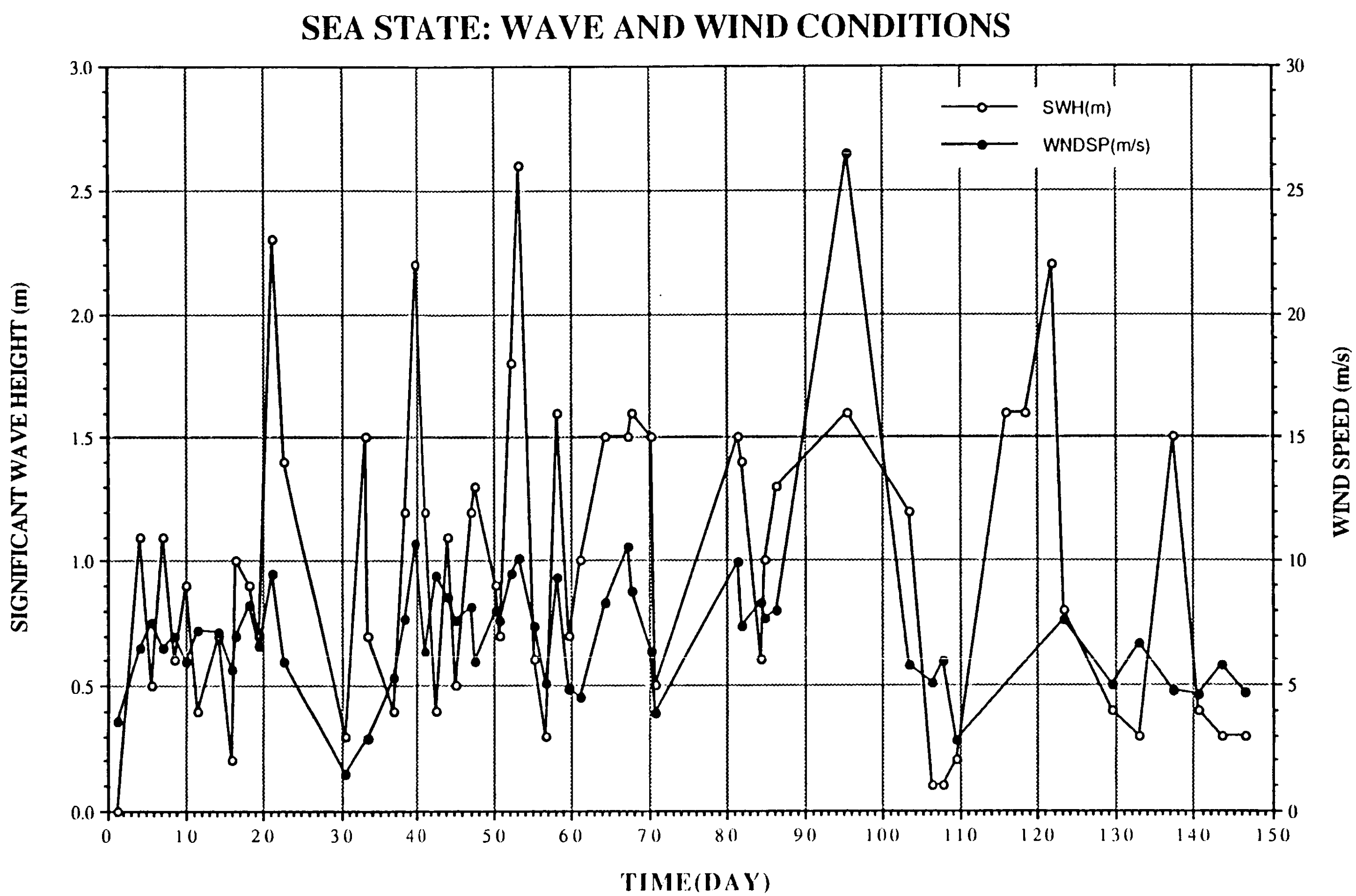
500-MILLIBAR HEIGHT CONTOURS AT 7:00 A.M., E.S.T.

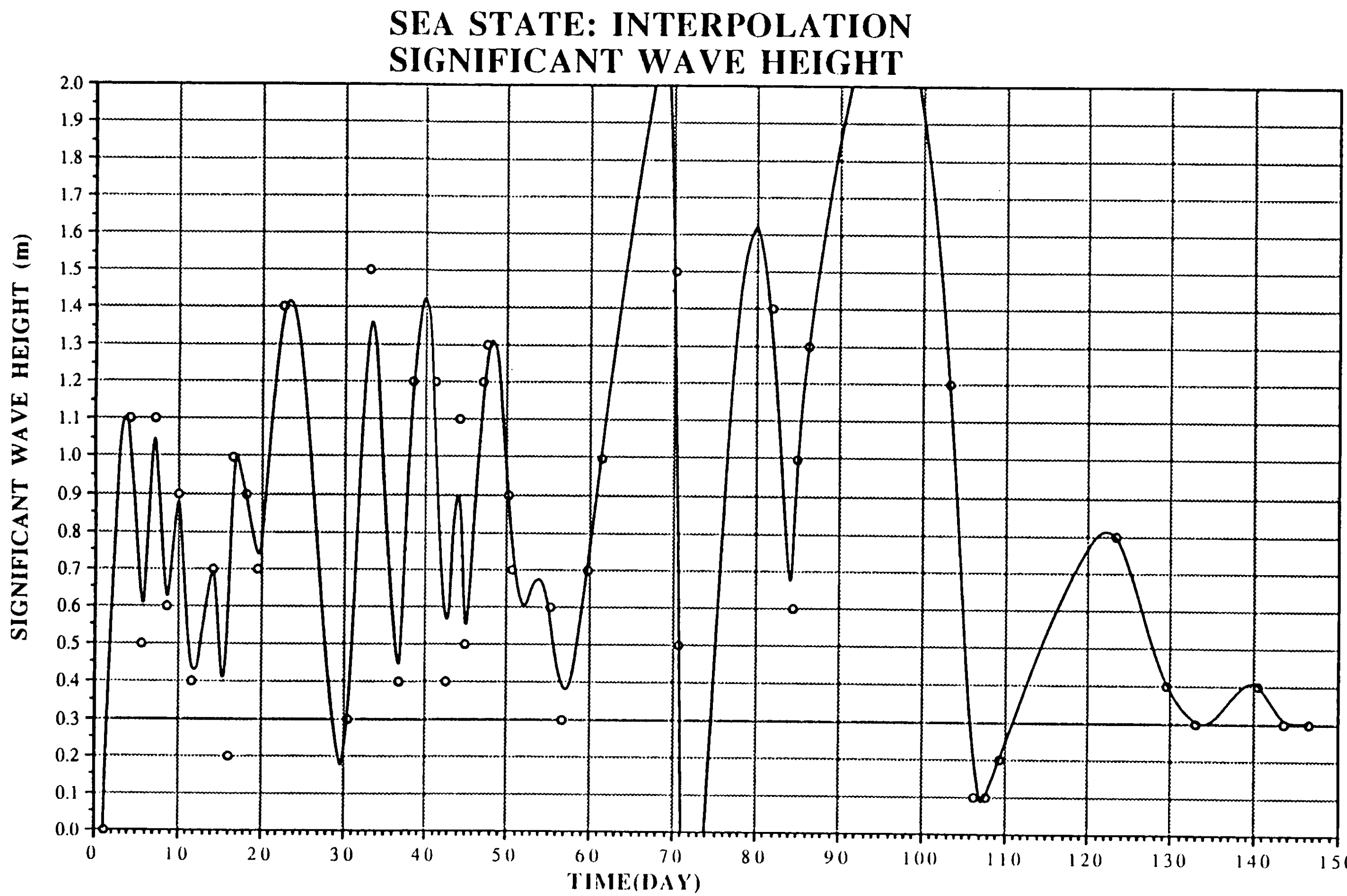

HIGHEST AND LOWEST TEMPERATURES


PRECIPITATION AREAS AND AMOUNTS

# APPENDIX I

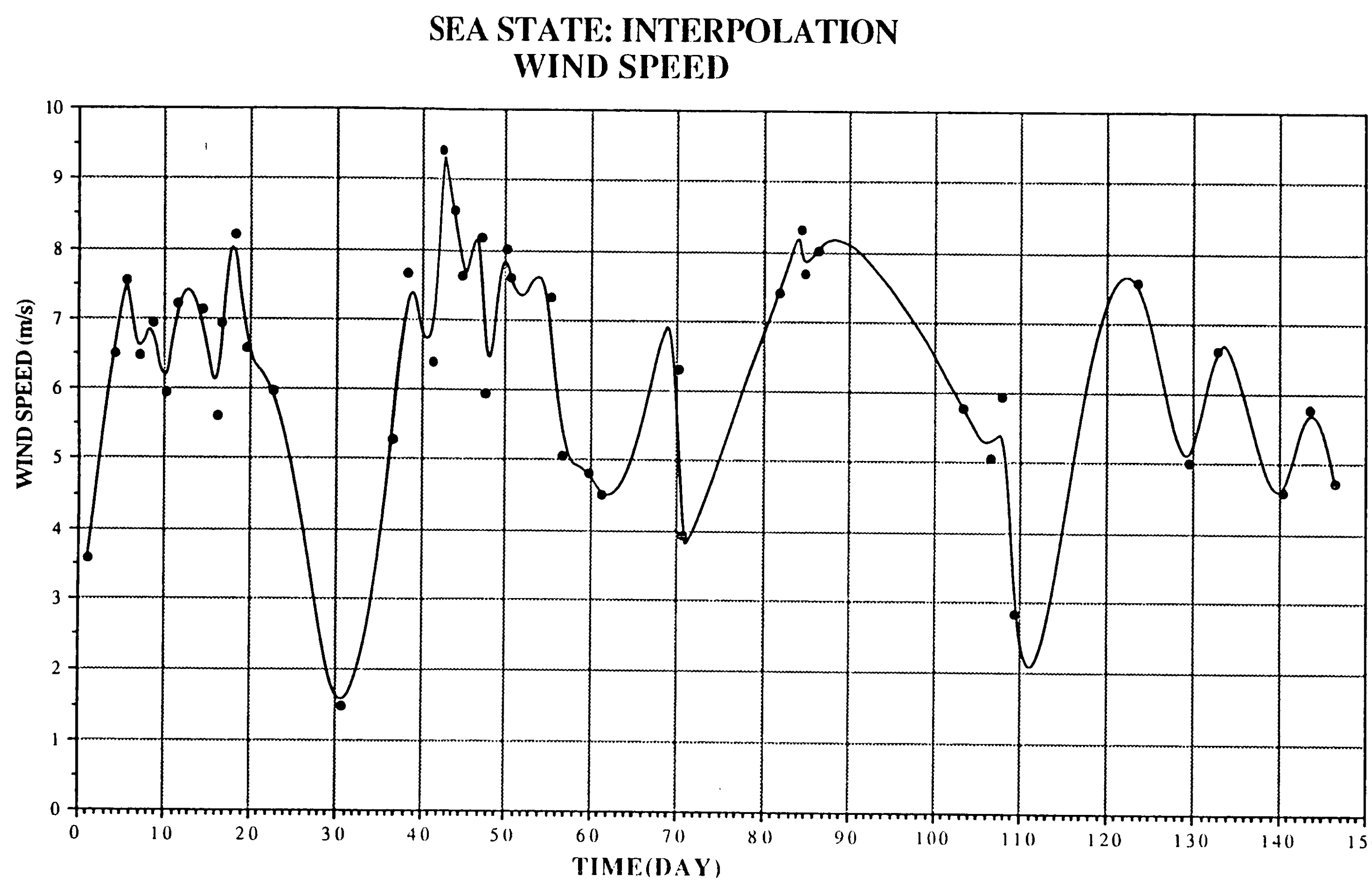
# HISTOGRAMS


SEA STATE: WAVE AND WIND CONDITIONS
SWH(m)
WNDSP(m/s)
SIGNIFICANT WAVE HEIGHT (m)
WIND SPEED (m/s)
TIME(DAY)



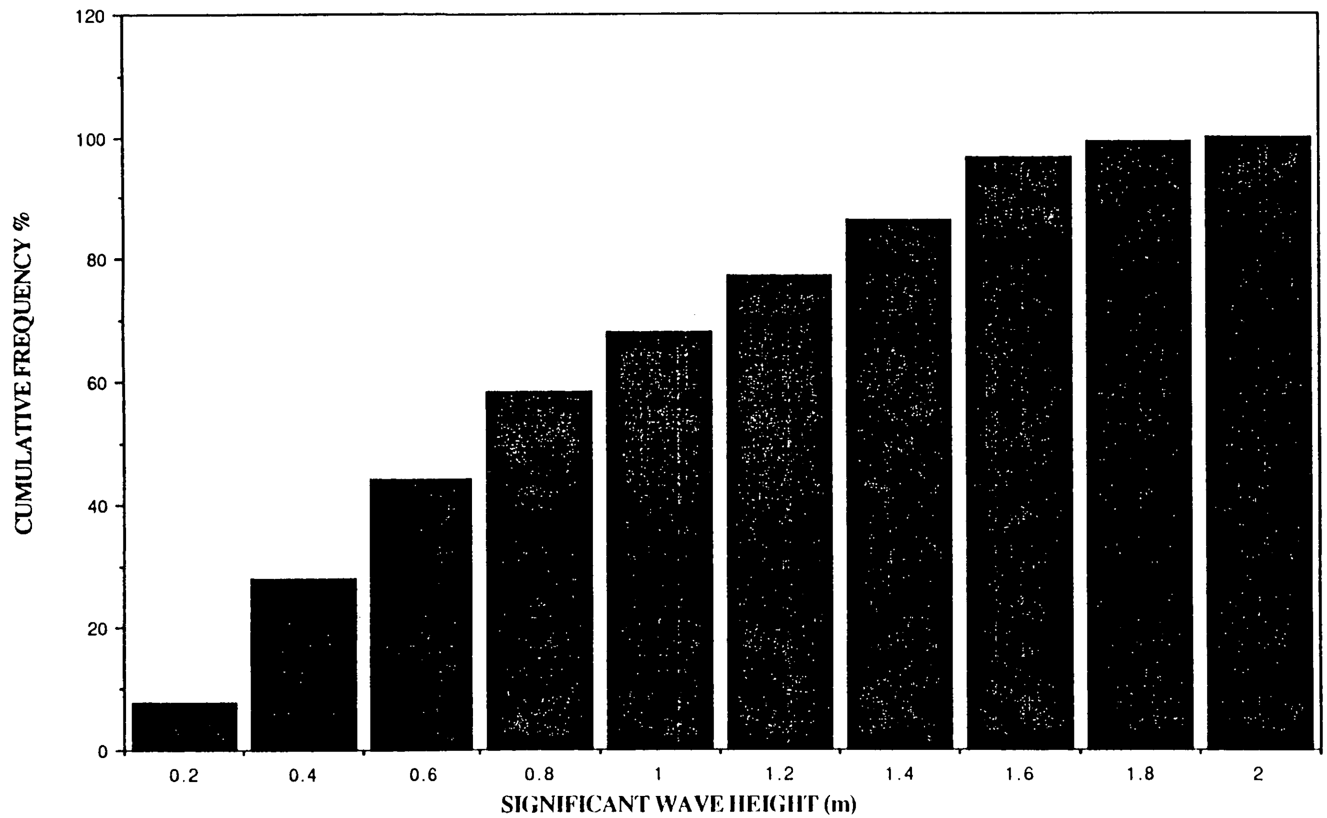
SEA STATE: INTERPOLATION
SIGNIFICANT WAVE HEIGHT
SIGNIFICANT WAVE HEIGHT (m)
TIME(DAY)
2.0
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150


I2


SEA STATE: INTERPOLATION
WIND SPEED
WIND SPEED (m/s)
TIME(DAY)
0
1
2
3
4
5
6
7
8
9
10
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150

I3

CALM CONDITIONS
SIGNIFICANT WAVE HEIGHT




I4

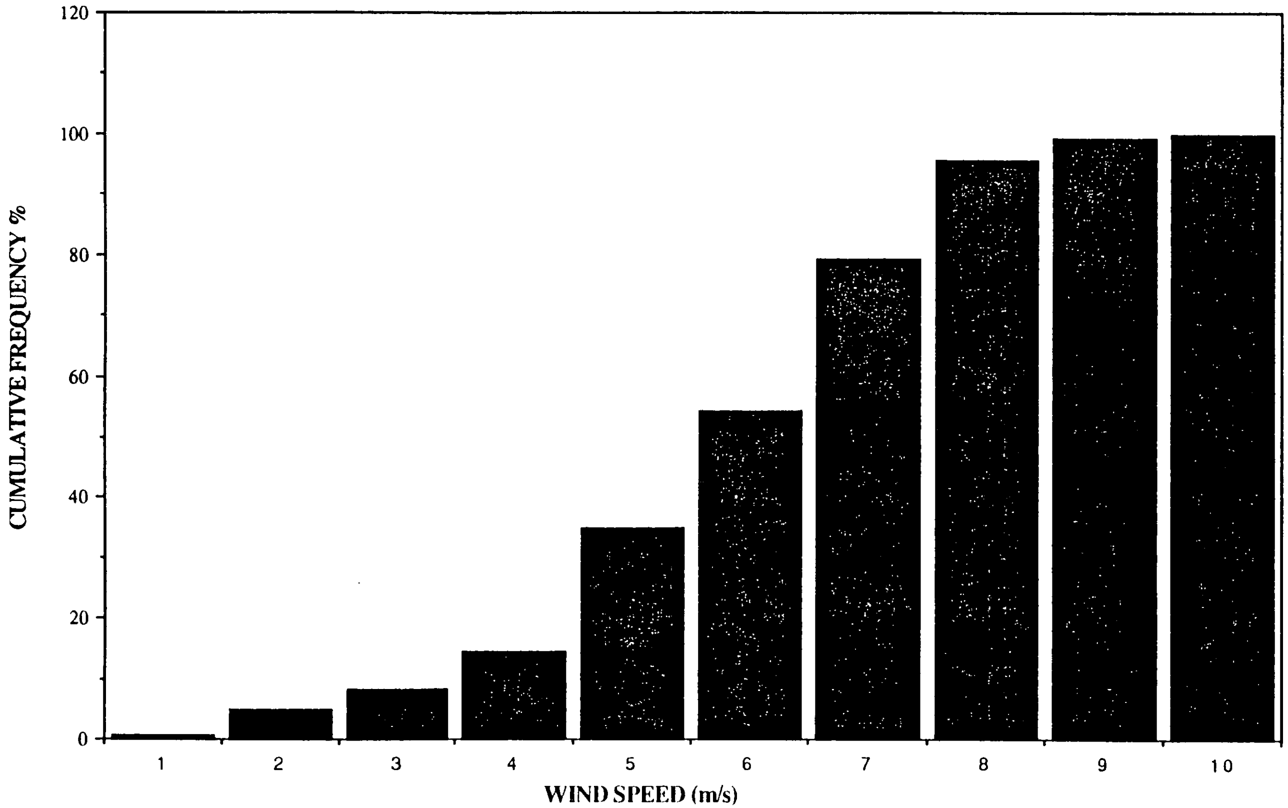
I5

SWHCALMDATA

Wed, Apr 4, 1990 15:39

| | SWH (m) | FREQ % | CUM % |
|---|---|---|---|
| 1 | 0.200 | 7.792 | 7.800 |
| 2 | 0.400 | 20.130 | 27.900 |
| 3 | 0.600 | 16.234 | 44.200 |
| 4 | 0.800 | 14.286 | 58.400 |
| 5 | 1.000 | 9.740 | 68.200 |
| 6 | 1.200 | 9.091 | 77.300 |
| 7 | 1.400 | 9.091 | 86.400 |
| 8 | 1.600 | 10.390 | 96.800 |
| 9 | 1.800 | 2.597 | 99.400 |
| 10 | 2.000 | 0.649 | 100.000 |

## CALM CONDITIONS
## WIND SPEED




I6

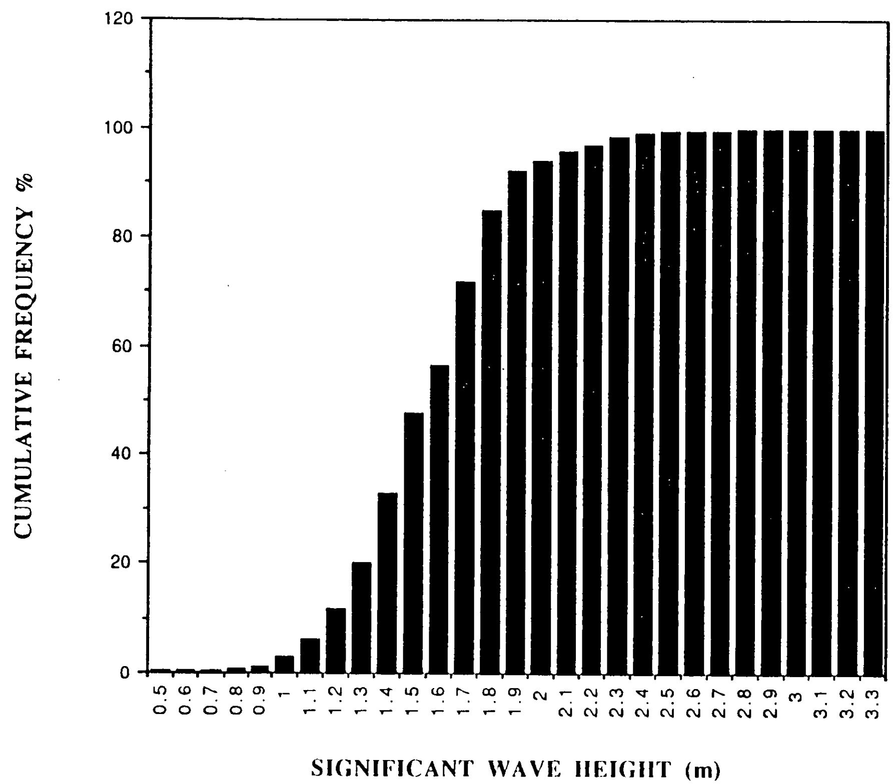
I7

WNDCALMDATA

Wed, Apr 4, 1990 15:42

| | WNDSP(m/s) | FREQ% | CUM % |
|---|---|---|---|
| 1 | 1.000 | 0.694 | 0.700 |
| 2 | 2.000 | 4.167 | 4.900 |
| 3 | 3.000 | 3.472 | 8.300 |
| 4 | 4.000 | 6.250 | 14.600 |
| 5 | 5.000 | 20.139 | 34.700 |
| 6 | 6.000 | 19.444 | 54.200 |
| 7 | 7.000 | 25.000 | 79.200 |
| 8 | 8.000 | 16.667 | 95.800 |
| 9 | 9.000 | 3.472 | 99.300 |
| 10 | 10.000 | 0.694 | 100.000 |

## CLASS 1 STORM CONDITIONS
## SIGNIFICANT WAVE HEIGHT




I8

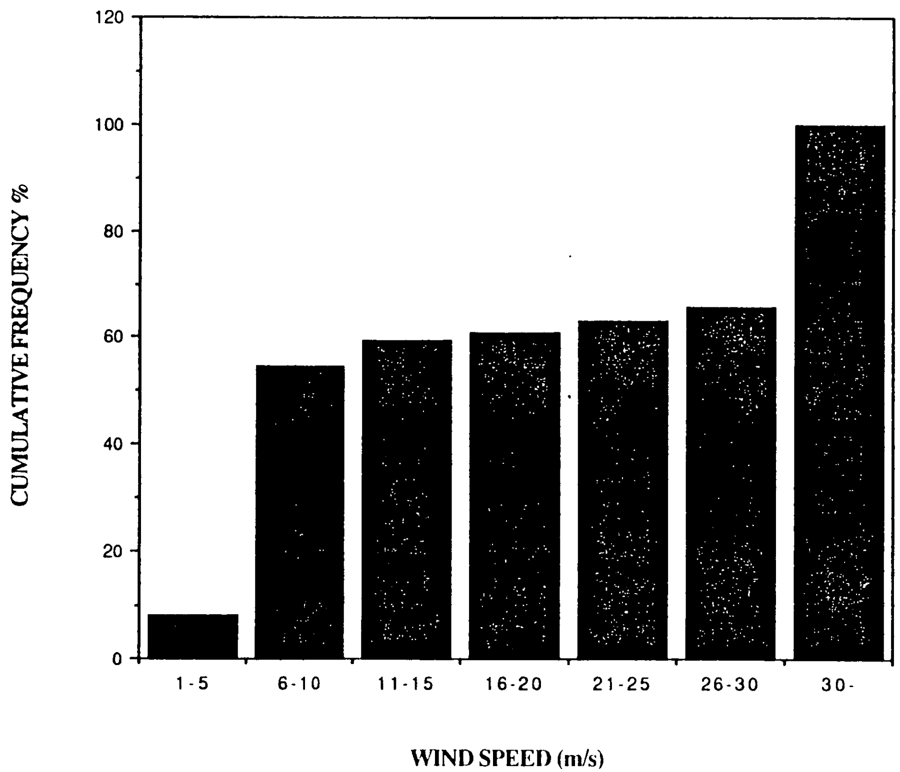
I9

STRM1SWHDATA

Wed, Apr 4, 1990 15:36

| | SWH(m) | FREQ % | CUM % |
|---|---|---|---|
| 1 | 0.500 | 0.204 | 0.200 |
| 2 | 0.600 | 0.000 | 0.200 |
| 3 | 0.700 | 0.088 | 0.300 |
| 4 | 0.800 | 0.292 | 0.600 |
| 5 | 0.900 | 0.563 | 1.100 |
| 6 | 1.000 | 1.965 | 3.100 |
| 7 | 1.100 | 3.174 | 6.300 |
| 8 | 1.200 | 5.302 | 11.600 |
| 9 | 1.300 | 8.342 | 19.900 |
| 10 | 1.400 | 13.020 | 32.900 |
| 11 | 1.500 | 14.934 | 47.900 |
| 12 | 1.600 | 8.515 | 56.400 |
| 13 | 1.700 | 15.371 | 71.800 |
| 14 | 1.800 | 13.307 | 85.100 |
| 15 | 1.900 | 7.127 | 92.200 |
| 16 | 2.000 | 1.980 | 94.200 |
| 17 | 2.100 | 1.702 | 95.900 |
| 18 | 2.200 | 1.318 | 97.200 |
| 19 | 2.300 | 1.157 | 98.400 |
| 20 | 2.400 | 0.713 | 99.100 |
| 21 | 2.500 | 0.348 | 99.400 |
| 22 | 2.600 | 0.260 | 99.700 |
| 23 | 2.700 | 0.000 | 99.700 |
| 24 | 2.800 | 0.130 | 99.800 |
| 25 | 2.900 | 0.000 | 99.800 |
| 26 | 3.000 | 0.093 | 99.900 |
| 27 | 3.100 | 0.093 | 100.000 |
| 28 | 3.200 | 0.000 | 100.000 |
| 29 | 3.300 | 0.000 | 100.000 |

## CLASS 1 STORM CONDITIONS WIND SPEED



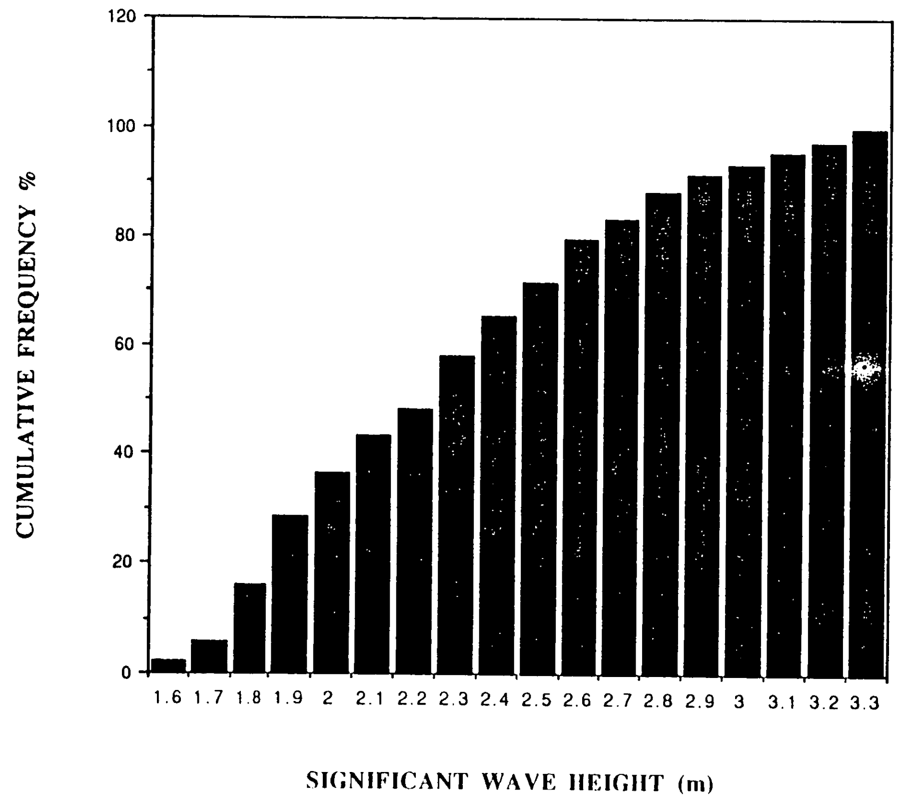

I 10

I11

STRM1WNDDATA Wed. Apr 4, 1990 15:33

| | WNDSP (m/s) | FREQ % | CUM % |
|---|---|---|---|
| 1 | 1-5 | 8.000 | 8.000 |
| 2 | 6-10 | 46.800 | 54.700 |
| 3 | 11-15 | 4.900 | 59.600 |
| 4 | 16-20 | 1.500 | 61.100 |
| 5 | 21-25 | 1.900 | 63.000 |
| 6 | 26-30 | 2.900 | 65.900 |
| 7 | 30- | 34.100 | 100.000 |

## CLASS 2 STORM CONDITIONS
## SIGNIFICANT WAVE HEIGHT




I12

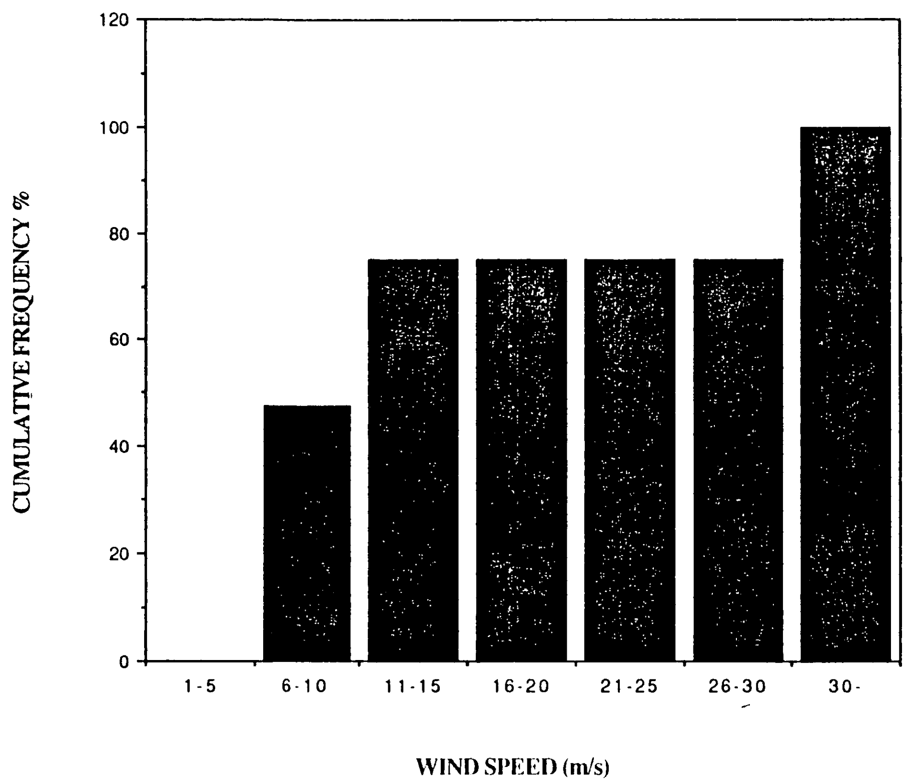
I13

STRM2SWHDATA

Wed, Apr 4, 1990 15:27

| | SWH (m) | FREQ % | CUM % |
|---|---|---|---|
| 1 | 1.600 | 2.200 | 2.200 |
| 2 | 1.700 | 3.600 | 5.800 |
| 3 | 1.800 | 10.100 | 15.900 |
| 4 | 1.900 | 12.700 | 28.600 |
| 5 | 2.000 | 7.700 | 36.300 |
| 6 | 2.100 | 7.000 | 43.300 |
| 7 | 2.200 | 4.900 | 48.200 |
| 8 | 2.300 | 9.800 | 58.000 |
| 9 | 2.400 | 7.400 | 65.400 |
| 10 | 2.500 | 6.200 | 71.600 |
| 11 | 2.600 | 8.000 | 79.600 |
| 12 | 2.700 | 3.500 | 83.100 |
| 13 | 2.800 | 5.100 | 88.200 |
| 14 | 2.900 | 3.400 | 91.600 |
| 15 | 3.000 | 1.700 | 93.300 |
| 16 | 3.100 | 2.200 | 95.500 |
| 17 | 3.200 | 1.800 | 97.300 |
| 18 | 3.300 | 2.700 | 100.000 |

## CLASS 2 STORM CONDITIONS WIND SPEED




I 14

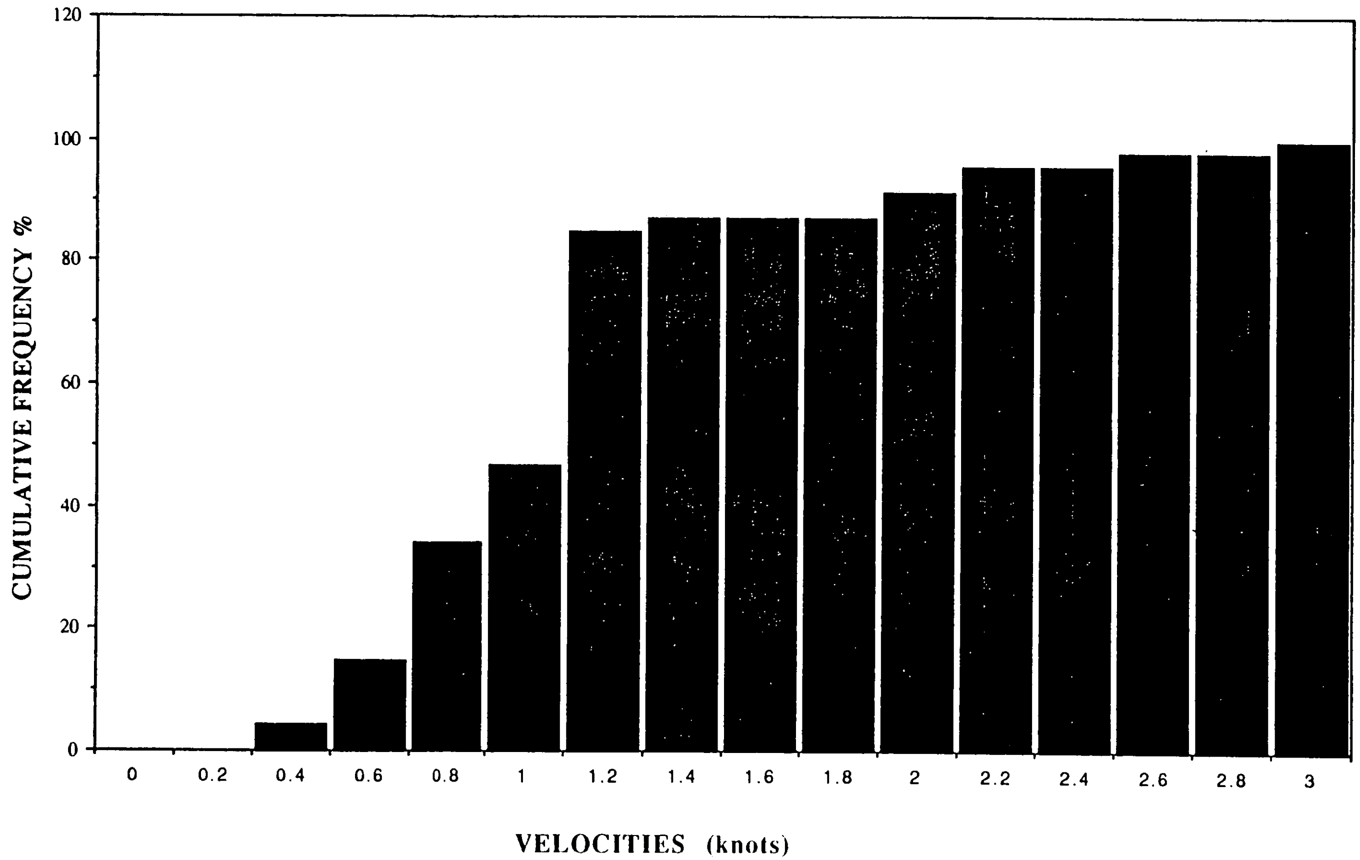
I15

STRM2WNDDATA

Wed, Apr 4, 1990 15:30

| | WNDSP(m/s) | FREQ % | CUM % |
|---|---|---|---|
| 1 | 1-5 | 0.000 | 0.000 |
| 2 | 6-10 | 47.600 | 47.600 |
| 3 | 11-15 | 27.400 | 75.000 |
| 4 | 16-20 | 0.000 | 75.000 |
| 5 | 21-25 | 0.000 | 75.000 |
| 6 | 26-30 | 0.000 | 75.000 |
| 7 | 30- | 25.000 | 100.000 |

# LOOP CURRENT VELOCITIES




I16

# APPENDIX J

# GLOSSARY

# APPENDIX J: GLOSSARY

Altimeter - measures distance between the satellite and the ground by measuring the time for a radar pulse to travel to the ground and back to the satellite.

AR - "autoregressive" time series forecasting technique that assigns weight to previous terms in the time series.

ARMA - "autoregressive moving average" combination of AR and MA models.

AVHRR - Advanced Very High Resolution Radiometer

Box - Jenkins - time series forecasting technique, often using ARMA models.

Eddy - large area of rotating water created by the passing of the Loop Current.

Fibonacci Sequence - numerical sequence in which each value is the sum of the previous two values.

GEOSAT - geological satellite launched in 1986 and was shut down in January 1990.

Heave - vertical motion of a ship in response to waves.

Loop Current - current which moves warm equatorial waters into the Gulf of Mexico west of Cuba and out between Cuba and Florida.

MA - "moving average" time series forecasting technique that assigns weight to current and previous random inputs.

Macroscopic Sea Height - level of the ocean surface relative to a fixed reference such as the center of the earth.

NASA - National Aeronautics and Space Administration.

NOAA - National Oceanic and Atmospheric Administration.

Radiometer - measures the radiant energy emitted by the earth's surface.

RAO - response amplitude operator, ships heave response to waves in distance of heave per unit wave height (ft/ft).

Significant Wave Height - distance from trough to peak of a wave.

SLAM - a dedicated simulation language which includes many helpful time keeping features.

Station Keeping - operations necessary to keep a dynamically positioned vessel stationary in the ocean conditions encountered.

USRA - University Space Research Association.